SYNTHETIC DATA GENERATION FOR TRAINING AND EVALUATION OF DEEP LEARNING-BASED COMPUTER VISION MODELS

DERİN ÖĞRENME-BAZLI BİLGİSAYARLI GÖRE MODELLERİNİN EĞİTİMİ VE DEĞERLENDİRİLMESİ İÇİN SENTETİK VERİ ÜRETİMİ

ABDULRAHMAN KERİM

ASST. PROF. DR. UFUK ÇELİKCAN

Supervisor

Submitted to

Institute of Sciences of Hacettepe University

as a Partial Fulfillment to the Requirements

for the Award of the Degree of Master of Science

in Computer Engineering.

June 2021

ABSTRACT

SYNTHETIC DATA GENERATION FOR TRAINING AND EVALUATION OF DEEP LEARNING-BASED COMPUTER VISION MODELS

Abdulrahman KERİM

Master of Science,**Computer Engineering Department**
Supervisor: Asst. Prof. Dr. Ufuk ÇELİKCAN
June 2021, 77 pages

The recent great success witnessed in computer vision field in solving high-level vision tasks such as visual object tracking, semantic segmentation, instance segmentation, and optical flow recognition is predominantly dependent on the availability of large-scale datasets, which are critical for training and testing new algorithms. Manually annotating visual data, however, is not only a time consuming process but also prone to errors and subject to privacy issues. In this work, we present NOVA, a general-purpose framework to create 3D virtual worlds populated with humans that provides pixel-level accurate ground truth annotations for many computer vision tasks. NOVA can simulate several environmental factors such as weather conditions or different times of day, and bring an exceptionally diverse and photo-realistic set of humans to life, each having a distinct appearance and features.

To demonstrate NOVA's capabilities, we utilized our framework to generate photo-realistic and diverse synthetic sequences for training and testing visual object tracking algorithms.

The main motivation was to show that the generated synthetic data, by our rendering engine, constitute a good proxy of its real-world counterpart and it can be deployed to boost the performance of learning based computer vision models. Particularly, our aim was to demonstrate the usability of our generated data for both training and testing computer vision models.

First, we generate two different synthetic datasets for the task of pedestrian tracking. The first of these datasets is utilized to assess the performance of some state-of-the-art visual trackers on various conditions. On the other hand, we employ the second one to train deep visual trackers to improve their performances on real sequences. Our study reveals that the tested trackers perform poorly in highly crowded scenes, or at low illumination and in foggy weather conditions. Additionally, the experiments demonstrate that our generated synthetic sequences indeed present a good proxy of the real sequences and it does improve the performances of deep visual trackers under standard and normal conditions. Following this, the essential question that emerged and required thorough experiments is the capability of our synthetic data to complement the real-world one and push the limits of current available visual object tracking datasets.

Bearing in mind the poor performance of the recent tracking algorithms at certain challenging conditions (as revealed by our previous experiments), we considered adverse weather conditions in more details. We provided a new person tracking dataset of real-world sequences (PTAW172Real) captured under foggy, rainy and snowy weather conditions to assess the performance of the current trackers. The considered trackers, both correlation filter -based or learning-based, showed a poor performance under these adverse weather conditions. Our experimental results link this deficiency to the lack of enough adverse weather training samples in the current visual object tracking datasets. To mitigate the problem, we extended our rendering engine to further simulate more realistic adverse weather conditions spanning foggy, rainy and snowy weather conditions. Pedestrians in rainy and snowy weathers are simulated with outdoor cold-weather clothes. Snow banks and water puddles are simulated to account for snow and water accumulations, respectively. Additionally, snow particles and rain drops are generated to match the videos in real life. In parallel to that, snow tracks left by

cars and pedestrians are simulated to give more realism. Pedestrians are randomly assigned umbrellas and the suitable animation is set accordingly. At the same time, fog is simulated using post-processing effects and the Enviro system. The severeness of each of the weather conditions is randomized at run time to give more diversity for the generated sequences.

Following this and harnessing the photo-realism and diversity of the simulated adverse weather condition, we provide a novel person tracking dataset of synthetic sequences (PTAW217Synth) generated by our NOVA framework spanning the same adverse weather conditions. The results demonstrated that the performances of the deep trackers under adverse weather conditions can be improved when our synthetically generated sequences are deployed for training.

Keywords: procedural content generation, synthetic-data for learning, rendering, visual tracking, person tracking

ÖZET

DERİN ÖĞRENME-BAZLI BİLGİSAYARLI GÖRE MODELLERİNİN EĞİTİMİ VE DEĞERLENDİRİLMESİ İÇİN SENTETİK VERİ ÜRETİMİ

Abdulrahman KERİM

Yüksek Lisans,Bilgisayar Mühendisliği
Tez Danışmanı: Yrd. Doç. Dr. Ufuk ÇELİKCAN
Haziran 2021, 77 sayfa

Bilgisayarla görme alanında görsel nesne izleme, anlamsal bölümleme, örnek bölümleme ve optik akış tanıma gibi üst düzey görme görevlerinin çözümünde tanık olunan son büyük başarı, büyük ölçüde eğitim için büyük ölçekli veri kümelerinin kullanılabilirliğine bağlıdır. Bu veri kümeleri yeni algoritmaları test edilmesi için kritik önem taşımaktadır. Bununla birlikte, görsel verilere el ile açıklama eklemek yalnızca zaman alan bir işlem değildir, aynı zamanda hatalara da açıktır ve gizlilik sorunlarına tabidir. Bu çalışmada, birçok değişik bilgisayar görme görevi için piksel düzeyinde gerçek değer ek açıklamaları sağlayan, insanlarla dolu 3B sanal dünyalar oluşturmak için genel amaçlı bir çerçeve olan NOVA'yı sunuyoruz. NOVA, hava koşulları veya günün farklı zamanları gibi çevresel faktörleri simüle edebilir ve her biri farklı bir görünüme ve özelliklere sahip, son derece çeşitli ve foto-gerçekçi bir insan grubunu hayata geçirebilir.

NOVA'nın yeteneklerini göstermek amacıyla, görsel nesne izleme algoritmalarını eğitmek ve test etmeamaçlı foto-gerçekçi ve çeşitli sentetik diziler oluşturduk. Ana motivasyonumuz, oluşturma motorumuz tarafından üretilen sentetik verilerin gerçek dünyadaki karşılığı için iyi bir alternatif olduğunu ve öğrenmeye dayalı bilgisayarla görme modellerinin performansını artırmak için kullanılabileceğini göstermekti. Özellikle amacımız, oluşturulan verilerimizin hem eğitim hem de bilgisayarla görme modellerinin test edilmesi için kullanılabilirliğini göstermekti.

İlk olarak, yaya takibi görevi için iki farklı sentetik veri kümesi oluşturuyoruz. Bu veri kümelerinden ilki, bazı son teknoloji görsel takip cihazlarının çeşitli koşullarda performansını değerlendirmek için kullanılır. Öte yandan, ikincisini, gerçek sekanslardaki performanslarını iyileştirmek için derin görsel izleyicileri eğitmek için kullanıyoruz. Çalışmamız, test edilen izleyicilerin çok kalabalık sahnelerde veya düşük aydınlatma ve sisli hava koşullarında kötü performans gösterdiğini ortaya koyuyor. Ek olarak, deneyler, oluşturduğumuz sentetik dizilerin gerçekten gerçek dizilerin iyi bir vekilini sunduğunu ve standart ve normal koşullar altında derin görsel izleyicilerin performanslarını iyileştirdiğini gösteriyor. Bunu takiben, ortaya çıkan ve kapsamlı deneyler gerektiren temel soru, sentetik verilerimizin gerçek dünyayı tamamlama ve mevcut görsel nesne izleme veri kümelerinin sınırlarını zorlama yeteneğidir.

Son izleme algoritmalarının belirli zorlu koşullarda (önceki deneylerimizin ortaya koyduğu gibi) zayıf performansı olduğunu akılda tutarak, bu alanı daha ayrıntılı olarak ele aldık. Mevcut izleyicilerin performansını değerlendirmek için sisli, yağmurlu ve karlı hava koşullarında yakalanan gerçek dünya sekanslarından (PTAW172Real) yeni bir kişi izleme veri kümesi sağladık. Hem korelasyon filtresi tabanlı hem de öğrenme tabanlı olan dikkate alınan izleyiciler, bu olumsuz hava koşulları altında zayıf bir performans gösterdi. Deneysel sonuçlarımız, bu eksikliği mevcut görsel nesne izleme veri kümelerinde yeterli olumsuz hava durumu eğitimi örneğinin olmamasına bağlamaktadır. Sorunu hafifletmek için, sisli, yağmurlu ve karlı hava koşullarını kapsayan daha gerçekçi olumsuz hava koşullarını daha fazla simüle etmek için oluşturma motorumuzu genişlettik. Yağmurlu ve karlı havalarda yayalar, soğuk hava kıyafetleri ile taklit edilir. Kar kümeleri ve su birikintileri sırasıyla kar ve su birikintilerini hesaba katacak şekilde simüle edilmiştir. Ek olarak, gerçek hayattaki videolara uyması

için kar parçacıkları ve yağmur damlaları oluşturulur. Buna paralel olarak, arabaların ve yayaların bıraktığı kar izleri simüle edilerek daha fazla gerçekçilik sağlanabilir. Yayalara rastgele şemsiyeler atanır ve uygun animasyon buna göre ayarlanır. Aynı zamanda işlem sonrası efektler ve Enviro sistemi kullanılarak sis simüle edilir. Hava koşullarının her birinin şiddeti, oluşturulan dizilere daha fazla çeşitlilik sağlamak için çalışma zamanında rastgele hale getirilir.

Bunu takiben ve simüle edilmiş olumsuz hava koşullarının fotoğraf gerçekçiliğinden ve çeşitliliğinden yararlanarak, aynı olumsuz hava koşullarını kapsayan NOVA çerçevemiz tarafından oluşturulan sentetik dizilerin (PTAW217Synth) izlediği yeni bir kişi veri kümesi sunuyoruz. Sonuçlar, olumsuz hava koşullarında derin izleyicilerin performanslarının, sentetik olarak oluşturulmuş dizilerimiz eğitim için devreye alındığında iyileştirilebileceğini gösterdi.

Anahtar Kelimeler: prosedürel içerik üretimi, öğrenme için sentetik veriler, işleme, görsel izleme, kişi takibi

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Asst. Prof. Dr. Ufuk Çelikcan for his enormous effort and support from the first day of this interesting journey. This work would never see the light without his motivation, passion, and consistent guidance. I would also thank Assoc. Prof. Dr. İbrahim Aykut Erdem and Assoc. Prof. Dr. Mehmet Erkut Erdem for the great effort, for their deep knowledge, and for the high quality research experience I learnt from them during this work.

I would also thank all members of COMPUTER GRAPHICS and GAME STUDIES LAB (HUCGLAB) at Hacettpe University. I would like to thank my colleagues for their enormous support and encouragement. Specially, my lab mate Alper Özkan with whom I discussed most of the interesting ideas I explored during my research.

I would like to add my special thanks to all my thesis committee members. Your suggestions, time, and effort are truthfully appreciated.

Finally, I sincerely appreciate my family help and support. I am specially and deeply grateful for my wife Emeni Kerim for her understanding, motivation, and for creating always the perfect environment and atmosphere for me.

This thesis was supported by TUBITAK-1001 Program (Grant No.217E029), GEBIP 2018 fellowship of Turkish Academy of Sciences awarded to E. Erdem, and BAGEP 2021 Award of the Science Academy awarded to A. Erdem

CONTENTS

Page

FIGURES

Page

TABLES

Chapter 1.

Introduction

The rapid progress in the field of computer vision and other AI related disciplines has been significantly driven by learning based methods, most notably those based on deep learning. Getting the best out of these approaches, however, broadly depends on the availability of large training data, and hence a major bottleneck on the way towards solving many computer vision tasks is the lack of diverse, accurate and large scale datasets. Manually curating such large datasets is labor-intensive and often error-prone. Although Amazon's Mechanical Turk or similar services can alleviate those issues, these tools are very expensive, especially for small research groups, if one wishes to capture the real-world in its full glory. But maybe more importantly, such crowdsourcing platforms become impractical for collecting ground truth data for some computer vision tasks (e.g. optical flow estimation). A neat idea to overcome these difficulties is to utilize synthetic data for machine learning, which has gained momentum over the past few years.

The large-scale benchmark datasets that were collected in the past few years [9–12] has lead to the unprecedented progress in deep learning based computer vision approaches. Although the exponential increase in the amount of digital data today can make data collection easier than before, manual labeling of large volumes of examples with high quality and accurate labels still requires too much effort and comes with a tremendous cost. Our proposed NOVA framework, with its procedural and automated generation capabilities, provides a solution to this daunting data collection/annotation challenge by letting the users create and render 3D virtual worlds containing human agents with different characteristics in real-time. The authors in [6] previously proposed a similar framework but their focus is mainly on human

action recognition and thus their framework has limited functionalities. On the other hand, in our proposed NOVA framework, the users have full control of the scenes, scene elements and humans, along with the illumination and weather conditions, allowing to study various factors affecting the success of their algorithms during development time and opening up a possibility being used in a wider range of computer vision tasks.

In the second part of the thesis work, person tracking in adverse weather conditions is considered in details. The adverse weather conditions are limited to snowy, rainy and foggy. A new real dataset called PTAW172Real is collected and annotated for assessing the performance of the trackers under these challenging conditions and we show their poor performance under such conditions. We link the degradation in performance to the lack of enough training sample under these conditions. We provide a solution by using our NOVA engine to generate a synthetic dataset, PTAW217Synth, that provides diverse and rich training sequences under adverse weather conditions. Our results show that using our synthetic sequences for training, we can boost trackers performance on real videos under these adverse conditions.

The recent improvements in game technologies have made the creation of photorealistic and physically accurate games possible. Since designing virtual worlds from scratch can be very expensive and requires highly skilled artists, it is possible to make use of the games that are already available. Making modifications on an open-sourced game or capturing the information sent by the game to graphics card can help to generate large synthetic datasets. However, the fact that commercial games do not represent a proxy of many real-world scenarios poses an essential problem with this approach, limiting its benefits.

Another way to create large synthetic datasets is to design the virtual world based on the needs. While it usually requires more effort to create and configure, this approach makes it possible to produce a high-fidelity proxy of the targeted scenarios. With the advances in graphics engine capabilities within the past decade, the photorealistic and physically-based simulations realized by using these engines allowed to minimize the gap between real and virtual world data.

Procedural generation has been proposed as a solution for creating realistic looking environments in relatively short amounts of time, making it easier and cheaper for users to generate virtual worlds from scratch. In its simplest form, a procedural generation framework follows some systematic recipes and generates scenes, populations and actions, based on the given set of instructions. Our work contributes to this line of research, in which we pay special

attention to the human generation aspect - in addition to offering a comprehensive variety of automatic ground truth annotation features that are partially available in other synthetic data generation frameworks.

Recently, convolutional neural networks (CNN) have shown a remarkable progress in various computer vision tasks such as object detection [13], object tracking [14], semantic segmentation [15], depth estimation [16], optical flow estimation [17], and person Re-Identification (ReID) [18]. While utilizing CNNs for computer vision field can improve both generalizability and accuracy, CNNs have an intrinsic restriction in terms of the data needed for training. Usually, better performance comes with deeper and larger CNNs which give a higher degree of non-linearity and more freedom in solving complex tasks. However, that introduces more variables for tuning. Unfortunately, training such models requires more data and more powerful computing devices. The introduction of cheap general purpose graphics processing units (GPGPUs) alleviated hardware limitation. However, the scarcity of large-scale datasets for training supervised learning methods remains as the main bottleneck for many computer vision tasks, especially, the ones that require enormous efforts for annotation, such as semantic segmentation and visual object tracking. Besides, for some others such as optical flow and depth estimation, it becomes extremely hard or even impossible to provide large-scale annotated datasets.

In addition to the need for large scale datasets, another requirement is a high level of diversity to allow deep learning models to work well in practice and not overfit to certain attributes. However, obtaining suitable datasets that are large and diverse from real world is not a simple task. Thus, small scale and mostly normal attributes tend to be the main features of the available datasets. Consequently, most of the available datasets tend to focus on normal scenarios under typical light conditions and camera parameters. The first reason behind this is the assumption that the computer vision model is going to be tested under these normal circumstances such as clear sky, optimal lighting, and standard recording conditions. While the second is the difficulty of obtaining datasets under rare conditions. Unfortunately, training computer vision models under these normal conditions causes unexpected behaviour or complete failure in adverse conditions.

Visual object tracking (VOT) is one of the major tasks in computer vision field that is essential for higher-level tasks such as pedestrian detection, action recognition, or trajectory estimation. Therefore, it is vital for many real-world systems such as self-driving vehicles,

automated retail or visual surveillance. Failure of such systems under adverse conditions can lead to property damages or human injuries.

In this work, we focus on person tracking under adverse weather conditions such as snowy, rainy and foggy weather conditions. Thereby, to assess the performance of the state-of-the-art trackers in person tracking in video feeds taken under such adverse conditions, we collect a novel real dataset, PTAW172Real, that consists of 172 videos featuring weather with heavy snow, rain or fog. Our experiments expose the poor performance of the state-of-the-art trackers when tested on PTAW172Real and this can be linked to the limited number of videos taken under adverse weather conditions in the current VOT datasets that these trackers were trained with. We offer a remedy for the lack of data availability by using our NOVA engine to generate a synthetic dataset, PTAW217Synth, that provides diverse and rich training sequences under adverse weather conditions. To the best of our knowledge, no work has been done to validate the usability of synthetic data for person tracking under adverse weather conditions. In this work, we show that using synthetic data, we can bridge the aforementioned gap and improve the performance of the learning-based trackers under adverse weather conditions.

Our main contributions in this work can be summarized as follows:

- We present a novel procedural content generation engine called NOVA. It is capable of generating large-scale and photo-realistic videos of human agents performing various actions on many different scenes along with the annotations for various computer vision tasks.
- Using our NOVA rendering engine, we generate two synthetic datasets specifically designed for person tracking. While we use the first dataset to assess the performance of existing visual trackers on various conditions, we employ the second one to train deep visual trackers to boost their performances on real sequences.
- Our experiments demonstrate that the existing trackers perform poorly in highly crowded scenes, or in scenes captured at night and in foggy weather conditions. Moreover, our generated synthetic sequences present a good proxy of the real sequences in that when used as training data, it improves the performances of deep visual trackers.

- We present a novel real dataset called PTAW172Real for visual object tracking under adverse weather conditions. The dataset contains 172 videos manually annotated covering snowy, rainy and foggy weather conditions.

- We highlight the poor performance of the state-of-the-art trackers under adverse weather conditions with PTAW172Real.

- Using our NOVA rendering engine, we procedurally generate a new dataset called PTAW217Synth made up of synthetic sequences under adverse weather conditions complete with automatically-generated per-frame annotations including bounding boxes at pixel-level accuracy, occlusion state and other relevant metadata such as time-of-day and camera type. The dataset consists of 217 sequences for person tracking spanning the three adverse weather conditions.

- We show that fine-tuning the pre-trained models on our synthetic dataset PTAW217Synth is able to improve the performance of the deep trackers. Similarly, we also show that training from scratch on only our synthetic training dataset can achieve comparable results to training on large scale real datasets.

Chapter 2.

Related Work

Creating realistic scenery, humans, actions and materials that mimic their actual world counterparts has been a major aim since the early days of video games. However, such a goal was not possible until recently. The ability to create photorealistic and physically accurate games motivated many researchers to investigate the possibility of utilizing them for the task of synthetic data generation. The works in this scope fall under either of the two main methodologies. The first is to adapt a specific game for the task of generating the synthetic dataset as in the works by Richter et al. [5, 19] where Grand Theft Auto V game was adapted to generate synthetic datasets. Essentially, they exploited the communication between the game and graphics hardware via injection of a middleware between the two to pull the necessary information for the desired annotations. Another work [20] modified Half-Life 2 game to evaluate a surveillance camera system. Using their proposed Object Video Virtual Video (OVVV) framework, they were able to generate bounding boxes and accurate segmentation labels for arbitrary number of frames automatically. In addition to that, they discussed how it is possible to integrate some noise and deformation techniques to produce more natural and realistic scenes. Similarly, [21] deployed a photorealistic video game to generate a large set of synthetic images, which were used to train a convolutional neural network for depth estimation and image segmentation. They concluded with many experiments that pre-training on synthetic data or training on both synthetic and real data achieve similar or better results compared to using only organic data for the training process. Nevertheless, using existing video games has the significant disadvantage of lacking diversity, as it does limit the number of scenarios, environments, actions, objects, and humans that can be included in a synthetic dataset.

The second methodology adopts using a graphics engine for data generation rather than individual video games. [22] used this concept by providing a plugin for Unreal Engine to generate ground truth for certain computer vision tasks by making some modifications on the internal data structures of a game and controlling a virtual camera to explore the scenes. Similarly, [23] used an open source driving simulator framework, VDrift, to generate a synthetic dataset, which incorporates high resolution images with their corresponding ground truth labels for semantic segmentation, depth and optical maps, specifically for multiclass image segmentation. A conditional random field model was trained with the synthetic data and used to analyze how various combinations of features affect the segmentation performance.

As an alternative, it is possible to refer to the open source animation movies to modify the rendering process to generate certain annotations along with the movie frames. One work [24] used this method for generating a synthetic optical flow dataset. They showed that optical flow statistics of their synthetic sequences and real video sequences are in agreement. Moreover, the dataset provided was larger than Middlebury [25] and KITTI [26] which allowed further studies on optical flow research. However, the inability to modify the scene structure of the animation constitutes the main drawback with this approach, making it even more limited for the purpose of synthetic data generation than using available photorealistic games.

Perhaps the most unrestricted way of creating arbitrarily large datasets together with their automated ground truth labels is taking the approach of using a graphics engine further by making use of procedural generation techniques in virtual world creation. De Souza et al. [6] investigated the possibility of adapting this concept with ragdoll physics, random perturbations and muscle weakening to generate a wide range of human actions systematically with their corresponding labels. They have defined 17 actions and showed that integrating the real-world data with their generated synthetic data can enhance the recognition performance. Another work [27] applied the concept of procedural generation to generate labeled crowd videos. As a proof of concept, it was shown that integrating their generated synthetic data with real-world data can improve the crowd behavior classifier's accuracy and the overall performance of pedestrian detection noticeably. Wrenninge et al. [28] demonstrated a photorealistic and diverse synthetic dataset that can be generated entirely procedurally. The ability to parameterize the scene generation process and the fact that these parameters are not correlated are the main contributions of this work. They showed that training on their

synthetic dataset and fine-tuning on organic dataset gives better performance compared to training only on the latter one only.

Due to the advancements in real-time rendering, the number of synthetic datasets that can be used for a wide spectrum of computer vision tasks has seen a considerable boost in the recent years. PHAV (Procedural Human Action Videos) [6] dataset is an example of a large scale synthetic dataset that was generated procedurally. It is mainly proposed for action recognition, and contains around 6 million frames in total. LCrowdV (Labeled Crowd Video) [27] dataset, which was produced by applying procedural modeling and rendering techniques, can be used for tasks such as pedestrian count, flow estimation and object detection and has more than 20 millions frames. On the other hand, there is VKITTI (Virtual KITTI) [8] dataset of approximately 21 thousand frames which can be used for multi-object tracking, scene level and instance level semantic segmentation and depth estimation in addition to object detection and optical flow estimation. SYNTHIA (Synthetic Collection of Imagery and Annotations) dataset [7], with more than 200 thousand images, is purposed for semantic segmentation and scene understanding of outdoor scenes for autonomous driving tasks. However, being specially designed for driving scenarios makes it inapplicable for many other computer vision tasks. Another similar and recent dataset is ParallelEye [29] which was generated by taking images from a synthetic car moving in a virtual city and contains around 40 thousand frames. It can be used for several tasks such as object detection, semantic and instance segmentation, and optical flow.

As discussed above, using computed generated imagery has become an important research direction especially for data-hungry deep learning approaches. That being said, the existing frameworks have some drawbacks. For instance, the main limitation of the frameworks proposed in [5, 19, 24] is that they do not allow to configure the virtual environments as they use existing computer games or computer generated movies while generating annotations for synthetic data. NOVA framework, on the other hand, lets the user to play with the environment along with the environment conditions such as weather, time of day, crowdedness and camera types. Moreover, including new features like new environments, new objects, or new character animations can be easily done due to its flexible design that supports procedural generation as opposed to the tools such as UnrealCV [22] which is just a plug-in for the Unreal game engine or the frameworks such as VDrift [23] that only supports driving based scenarios.

Another advantage of NOVA lies in the annotations it supports. As compared to the frameworks suggested in [20, 21], NOVA allows to extract a richer set of annotations for a user generated scene. These include accurate annotations for some low-level vision tasks such as scene depth, optical flow and surface maps, and annotations for some high-level tasks such as object detection, visual tracking, semantic segmentation and instance segmentation. Besides, from the human agents perspective, our main focus is not the human action recognition as in [6] or crowd behavior learning and counting in [27]. With the capability of procedurally generating a large and diverse set of synthetic humans and their character animations, it suggests a more generic solution which opens many possible applications.

With the proposed NOVA framework, our main aim is to further advance the efforts in computer vision by facilitating the automated creation of new arbitrarily large synthetic datasets with an extensive variety of ground truth annotations. NOVA lets users easily create photo-realistic 3D virtual worlds containing procedurally generated humans, and allows to obtain frame and pixel-level annotations about a scene and its elements in real-time, making it a versatile framework for automatic data collection and labeling pipeline for a wide range of tasks including but not limited to visual tracking, crowd counting, semantic segmentation, optical flow estimation, and depth estimation. It can simulate several illumination and weather conditions such as fog, rain, snow, daytime, nighttime, which help to test both favorable and adverse settings for these tasks. Furthermore, procedural generation capabilities of NOVA allows to generate unique synthetic humans with very diverse characteristics regarding body shape, gender, age and clothing, making NOVA a perfect tool for generating realistic-looking synthetic data for problems involving persons.

Despite the fact that deploying synthetic data in computer vision field has just started recently, a number of works investigated the usability of synthetic data for different computer vision tasks. In general, synthetic data can be employed for both training and testing purposes. For training, they can be used as the only training data, or to augment the real ones. It is possible to apply synthetic data for pre-training or fine-tuning learning models as well.

One work [30] investigated the usability of synthetic data for instance segmentation and object detection. They concluded that training on both synthetic and real data achieves better results as compared to training on a small set of real data. At the same time, they show that fine-tuning on their augmented data can achieve even better results. Similarly, Cheung et al. [27] proved that synthetic data can be used with real data to improve accuracy for crowded

scene understanding. They show that using their generated synthetic dataset, LCrowdV, with real datasets can improve the accuracy as compared to using these real datasets alone.

Varol et al. [31] demonstrated the usability of synthetic data for human depth estimation and part segmentation. They prove that training on synthetic and real images increases the accuracy for semantic segmentation and reduces the root-mean-squared-error for depth estimation. In the same way, Barbosa et al. [32] extensively studied the advantages of using their generated synthetic dataset, SOMAset, for the task of person ReID. They show that pre-training on their synthetic dataset then fine-tuning on real datasets achieves better results as compared to training only on real datasets.

Under the scope of visual object tracking, Gaidon et al. [8] provided a detailed analysis on the advantages of using synthetic data for the task of multi-object tracking. They show that training on their synthetic dataset then fine-tuning on real datasets achieves the best results as compared to only training on synthetic or real datasets.

Similarly, Zhang et al. [33] used image-to-image translation method to generate synthetic thermal infrared tracking videos using the RGB ones. They show that training on their synthetic videos then fine-tuning on real ones or training on both synthetic and real videos achieve better results as compared to training on the available small scale real datasets.

Similar to the previously mentioned works, we also investigate the advantages of using synthetic data for training learning-based models. However, this work sheds light on the limitations of the available real and synthetic visual object tracking datasets. As shown in Fig. 4.1., the adverse weather conditions seem to be underrepresented in most of the available real and synthetic VOT datasets. This causes the state-of-the-art trackers perform poorly under these challenging weather conditions. Bearing this in mind, we present synthetic data as a legitimate solution for the lack of the adverse weather conditions in the real datasets. To this end, we utilize our procedural content generation engine NOVA to generate a visual object tracking dataset to be used in the training of general purpose visual object trackers. The generated dataset is specifically designed for tracking people under adverse weather conditions in outdoor environments.

Chapter 3.

NOVA Rendering Engine

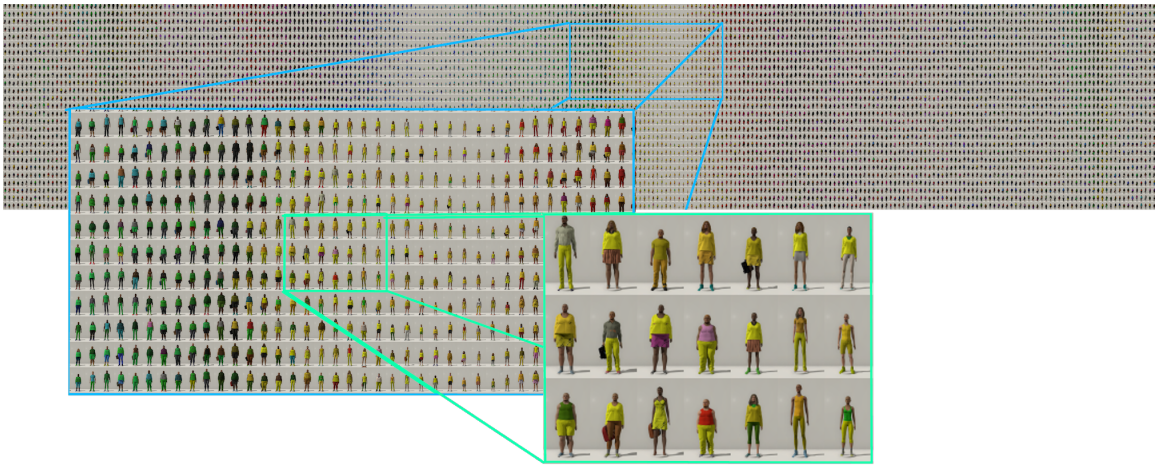


FIGURE 3.1.: A sample of 21 synthetic humans (in focus) from a set containing 9112 unique humans generated by NOVA.

3.1. NOVA: Framework of Rendering Virtual Worlds with People for Computer Vision Tasks

Cem Aslan did the procedural generation of synthetic humans that are based on UMA and the ground-truth generation parts. All the remaining aspects of the rendering engine were done in the scope of this work.

Algorithm 1: Algorithm for Synthetic Humans Spawning

$SPs \leftarrow$ Current Scene Predefined Spawning Points;
$Spawn_{center} \leftarrow$ Get Generated Camera Position;
$Spawn_{radius} \leftarrow$ Get Pedestrians Sparsity Value;
foreach *SP* $\in$ *SPs* **do**
    if *SP is within the camera view volume* **then**
        $SP.dist2Cam \leftarrow$ Get Current Spawning Point Distance to Camera;
        if $SP.dist2Cam < Spawn_{radius}$ **then**
            Activate SP;
$Peds \leftarrow$ Get All Pedestrians;
foreach *Ped* $\in$ *Peds* **do**
    $SP_{random} \leftarrow$ Randomly Pick an activated SP;
    $Ped.position \leftarrow SP_{random}.position$;

Our framework NOVA is built on the widely used Unity graphics engine. The framework, when all annotations are enabled (except bounding boxes, which are computed offline) and the number of synthetic humans to be generated is set to vary between 5 and 15, runs at real-time speeds (rendering between 42 and 60 frames per second on average) using current generation hardware (Intel Core i7-7700HQ, GeForce GTX 1070, with SSD and 32GB RAM). Readers are referred to visit the project website `https://graphics.cs.hacettepe.edu.tr/NOVA` for an online demo of the framework that allows to observe all procedural generation and visual ground-truth annotation features of NOVA at real-time by adjusting various scene-level attributes.

NOVA consists of the following data generation and annotation features to facilitate the creation of arbitrarily large datasets for a diverse array of computer vision tasks from pedestrian detection to scene understanding.

3.1..1 Humans

NOVA populates an environment with synthetic humans on a random selection of predefined spawning points that are within the view volume of the generated camera. A sparsity parameter is used to control the distribution of the spawning points, which determines the level of human crowdedness in the view as explained in Algorithm 1.

TABLE 3..1: Statistics about unique item variations in the procedural generation of synthetic humans. Possible variations in color are additionally provided inside parentheses.

Facial Items

Item	Male	Female
Hair	4 (48)	3 (32)
Eyebrows	2(24)	2 (24)
Beard	8 (96)	- / -

Clothing and Accessory Items

Item	Male	Female
Upper-Body Clothing	7 (28)	7 (28)
Lower-Body Clothing	6 (240)	13 (520)
Outerwear	2 (80)	3 (120)
Shoes	5 (40)	10 (80)
Bags	3 (12)	3 (12)
Other	2 (4)	3 (18)

The synthetic humans are procedurally generated at run-time by making use of several content creation layers which consist of a predefined set of categorizable, annotatable features as well as procedural, low-level randomizations to these features. The low-level randomizations further enhance the variations realized by the hand-tailored annotatable features in order to substantiate uniqueness in generated humans in arbitrarily large sets (Fig. 3.1.). This population process is built upon the publicly available UMA system [34].

To procedurally generate a synthetic human, a unique body and face shape are first created from either male or female base meshes. The attribute set to morph the body mesh is calculated from a base set of pre-determined body attributes. For each gender, there are three sets of height types (*short, average, tall*), three sets of weight types (*thin, athletic, overweight*), and two sets of age types (*child, adult*) available. One from every attribute type is randomly selected and the values are blended together considering their effects on different morph points. For instance, a tall child, while being taller than the average of the children generated, would still be shorter than an average adult. Once a distinguishable and annotatable body type (e.g., '*short athletic adult male*') is realized from the blended attribute set, it is further randomized by applying a rather small white noise with uniform distribution to each morph point on the body in order to ensure uniqueness while still resembling the tagged body type. This process theoretically allows to create infinitely many unique bodies which can be categorized into 36 major body types [35].

Then, a set of clothes and facial attributes are generated for the synthetic human from a set of recipes, which create a content instance by mixing and recoloring several recipe items in unique ways (Table 3..1). For example, a recipe for creating a beard texture contains three options for beard masks which are randomly selected in varying numbers, blended together

(if more than one mask is selected) and used for applying a beard matched to the human's hair color, potentially generating eight different beard shapes. On the other hand, a recipe for choosing a shoe is relatively simple and selects one of the shoe meshes provided for the corresponding gender.

A shared color system is used for applying colors, such that, each recipe chooses a color from a set of different palettes for skin, hair and clothing types. These colors are then multiplied with one of the alternative mask textures in order to yield variety in hair and skin textures and clothing patterns. The resulting colored and patterned textures are then used as the diffuse channel of the material while others (specular channel, gloss channel, etc.) are kept unchanged in order to retain correct physically-based material properties. This recoloring scheme allows us to further diversify the created humans while still keeping an easily categorizable generation system.

The resulting meshes from the recipe-based generation process are skinned onto the skeleton with the body mesh and the additional texture masks which are used to cull the body parts that will be covered by these meshes are added onto the base mesh textures during sampling. Fig. 3.1. shows an arbitrarily chosen subset of a sample of 9112 unique humans generated by NOVA. Although the instances in the figure are arranged with respect to perceptual similarity, it can be seen that even the humans in the small subset are still easily distinguishable from one another.

The animations for the humans are procedurally generated by blending between several motion captured animation sets including standing idle, walking, running and arguing. In order to create a unique motion instance at each time, two of these sets are randomly chosen and blended together. The blending is handled using linear interpolation, such that a blended animation is an average of the separate animations weighted randomly by uniformly distributed blending parameters. As the humans are created using a common rig structure that adapts automatically, each can be assigned a randomly blended animation with seamless instant mapping.

The employed motion sets are limited to the ones that are most commonly encountered within the compatible real-world datasets. Additional sets of motions can be easily incorporated into the framework to advance variety. It should be noted that the duration of the generated video sequences is not limited by the duration of the motion clips and NOVA can generate video sequences of arbitrary duration by looping the blended animations as needed.

The blended animations involving locomotion are kept consistent with the environment geometry by using Unity's navigation mesh system which facilitates path planning and obstacle avoidance along a path. The destination of a path is assigned randomly by NOVA and if the destination is reached before the sequence ends, a new one is assigned.

3.1..2 Environments

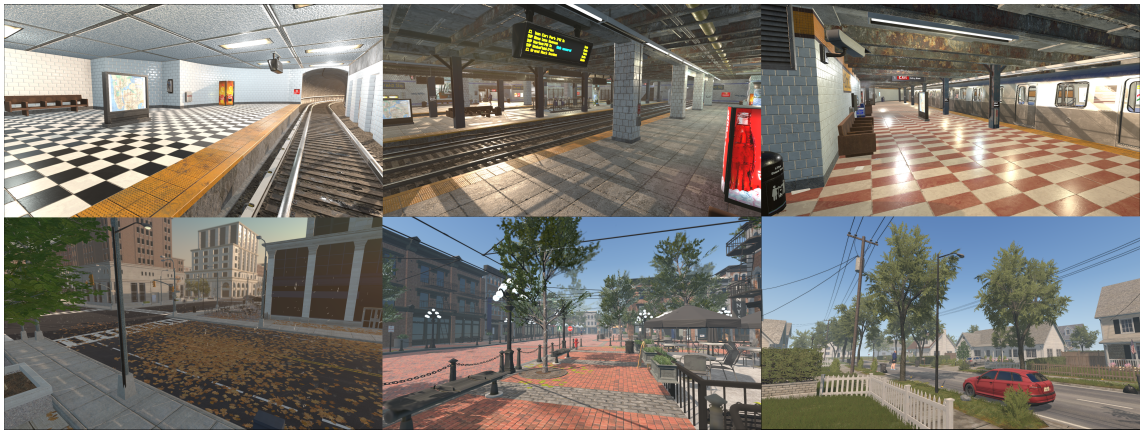
Currently, NOVA can create sequences in three outdoor environments (a town square, a suburban street and a metropolitan urban district) and one indoor environment (a subway station) (Fig. 3.2.(a)). Each environment is equipped with at least 20 different spawn points, which are selected at random during population process. Lighting in the 3D environments is parametrically generated to simulate different hours of a day (Fig. 3.2.(b)) and weather types based on sun direction and altitude (Fig. 3.2.(c)). The skybox, which provides ambient lighting for the 3D environments, and the weather effects are procedurally generated using the Enviro system [36].

Moreover, NOVA also makes use of HDR cubemaps that are captured from real-life (Fig. 3.2.(d)). In this case, the synthetic human receives directional lighting from the virtual sun and ambient lighting from the cubemap by using the image-based lighting method [37]. In order to blend the generated human with the environment further, the shadow that would be cast by the human on the ground is simulated by using a transparent plane, which receives shadow from the human's mesh. Although the background seems more realistic compared to the 3D environments, the drawback to using cubemaps is that illumination and weather changes can not be applied to them procedurally without ending up looking non-realistic in general.

3.1..3 Cameras

NOVA simulates different camera types as follows.

Surveillance Cameras: include both static and PTZ type surveillance cameras. The PTZ camera performs panning, tilting and zooming to keep the human being tracked in its field-of-view.



(a) Sample images of the 3D environments. First row: a subway station. Second row from left: a metropolitan urban district, a town square, and a suburban street.

(b) Different times of day

(c) Various weather conditions

(d) Samples using HDR cubemaps [38] captured from real-world

FIGURE 3.2.: Illustrating the diversity in NOVA's computer-rendered synthetic environments.

Algorithm 2: Algorithm for Non-Surveillance Camera Operation

Activate Camera Paths for the Specified Camera Type;
Set Camera Parameters;
$ID_{tracked} \leftarrow$ ID of the Synthetic Human Being Tracked;
$ID_{tracked}.Collider.Radius \leftarrow$ Higher Collider Radius Value than Others;
foreach *CameraPathCollider* $\in$ *Active Camera Path Colliders* **do**
    if *CameraPathCollider is triggered by* $ID_{tracked}.Collider$ **then**
        Set the Camera Attached to CameraPathCollider as the Active Camera;
        $ID_{tracked}.Collider.Radius \leftarrow$ Regular Collider Radius Value;
        Set the Active Camera to Follow and Look at the Object Rotating about $ID_{tracked}.Joints.Hip$;
while $ID_{tracked}$ *is occluded* **do**
    Wait;
Start Recording;

(a) RGB image (b) Bounding box (c) Body segmentation (d) Body pose (e) Textual annotations

FIGURE 3.3.: Sample of human-level annotations automatically generated for a synthetic human.

Non-Surveillance Cameras: include UAV and ground-level camera types. The first one simulates a camera attached to a UAV while the second one imitates a pedestrian carrying a camera and recording others. For each type, there is a predefined set of camera paths, which has a separate camera assigned per path, in each environment. The non-surveillance camera operation is outlined in Algorithm 2. To avoid having the tracked human always right in the middle of the view, the camera follows a virtual object rotating in an orbit around the human's hip instead of tracking the human directly.

3.1..4 Ground Truth Annotations

NOVA automatically generates ground-truth annotations on-the-fly as the simulated scene is procedurally created and photorealistically rendered for each frame. All annotations, except the textual metadata, are at the pixel-level.

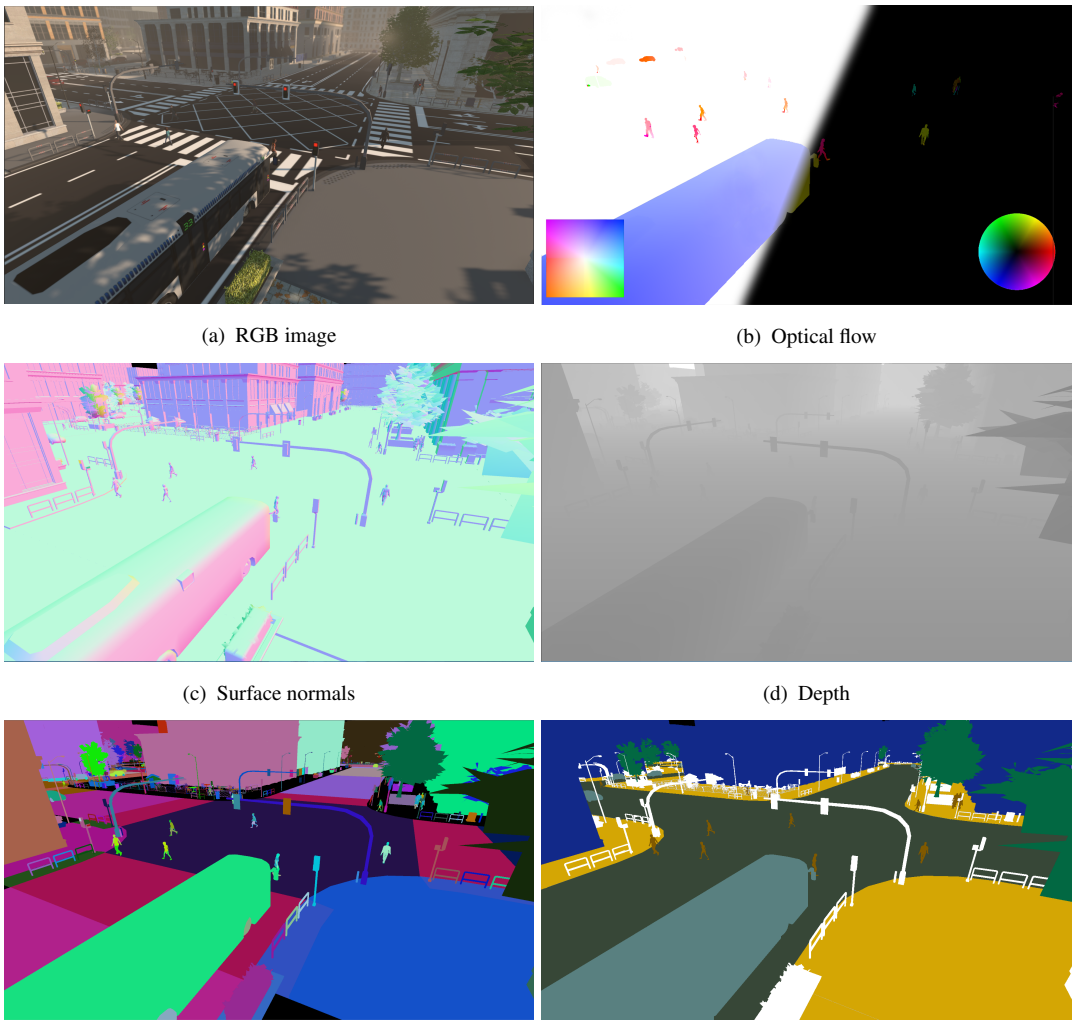
For each screen-space annotation, a separate camera is created and each camera uses different shaders, shader-specific parameters and culling parameters in order to create that annotation's frame. An effects shader containing sub-shaders for the annotations is set to each of these cameras as replacement shader which then uses the sub-shader with the matching render type of the specified annotation. That is, the camera renders the scene as it normally would, i.e., the objects still use their own materials, but the actual shader that ends up being used for annotation is changed, overriding shaders for regular rendering, and, instead, outputting the annotation.

Optical Flow. For the optical flow pass, the pixel motions are encoded in screen UV space to a screen-sized RG16 (16-bit float per channel) texture. Color encoding is done according to per-pixel motion vectors with respect to the camera. This information comes from an extra render pass into which moving objects are rendered and their motion is constructed with respect to inter-frame differences. Different optical flow annotation schemes can be applied by changing mappings for the encoding in order to make it compatible with existing datasets. Fig. 3.4.(b) exemplifies two such alternative encoding schemes. Optical flow sensitivity can be adjusted as desired so that the amount of movement that is to be observed is encoded in a normalized manner.

Surface Normals. During the surface normals pass, surfaces are color encoded according to their orientation with respect to the camera (Fig. 3.4.(c)). Encoding is done using stereographic projection into a 16 bit value which is packed into two 8 bit channels of a screen-sized texture. This information comes directly from the G-buffer.

Depth Map. For the depth map creation, pixels are gray-level indexed based on per-pixel distance to the camera (Fig. 3.4.(d)). The information for depth map textures comes directly from the actual depth buffer which is also a product of the G-buffer rendering.

Instance Segmentation. For every frame, each distinct entity within the camera view is assigned a unique identifier color representing its object ID (Fig. 3.4.(e)). The view is then



(a) RGB image (b) Optical flow

(c) Surface normals (d) Depth

(e) Instance segmentation (f) Semantic segmentation

FIGURE 3.4.: Sample of scene level annotations automatically generated by NOVA.

rendered by outputting the respective color without additional shading to obtain the instance segmentation pass.

Semantic Segmentation. Entities within the camera view are also assigned colors based on layers representing their category, e.g., human, vehicle, road (Fig. 3.4.(f)). The assigned colors are then rendered without additional shading to obtain the semantic segmentation pass. The variety of categories can be expanded as desired by defining additional layers. The layers should be assigned to the respective objects or their prefabs during the content creation process.

While creating instance segmentation and semantic segmentation frames, unique object identifiers and layers are encoded into RGB color values, set into a block of material values and passed into the replacement shaders [39] to be used. This process is repeated every time a change occurs in the scene, e.g. when a new human is generated.

NOVA can also provide the class and instance -level segmentation maps for which only a set of chosen objects are culled, e.g., to generate ground truth data for person tracking, everything except the synthetic humans in the frame are culled. These masked versions work in the same fashion as their non-masked counterparts but are rendered using a separate camera instance that only uses that set's layer for culling.

Bounding Box. For the bounding boxes, NOVA provides a segmentation that masks each human in view with a different color. This segmentation is used for min-max calculations to compute the per-frame bounding box for each human. Since this process takes considerably more time than the other annotations NOVA generates, especially for crowded simulations, the second step is carried out offline once all the other data is generated at real-time.

Body Part Segmentation. Body part segmentation of a synthetic human (Fig. 3.3.(c)) is generated by assigning separate vertex colors to each vertex for torso, head, arms and legs. For this, NOVA checks the bone weights of every vertex of a human mesh when it is first generated. Each vertex is assigned to one of the six colors for the respective body part depending on the weights of the bones that the vertex is connected to. The colors are then linearly interpolated during the fragment stage to achieve the final result. This process allows scalability as it can be carried only once when a synthetic human is first generated, allowing to keep using GPU for skinning with a higher frame rate during rendering.

Body Pose. To create the body pose information of a synthetic human in a frame, the positions of the skeletal joints are transferred into the screen-space and output as values normalized with respect to image size. In addition to the screen-space positions of the joints, NOVA also outputs a depth value per joint which can be used to resolve conflicts such as overlapping or occlusion. The output is in textual metadata form to allow flexibility in visualization. For instance, the body pose visualization in Fig. 3.3.(d) is compatible with the keypoint detection format of COCO dataset [10].

Other Textual Annotations. Some other attributes (see Fig. 3.3.(e)) of a generated human that are not suitable to be output as image modalities are output as textual metadata. Most of these attributes were chosen to reflect the ones which are present in existing datasets of real

TABLE 3..2: Distributions of attributes across the sequences in our synthetic person tracking dataset generated by using NOVA.

Attribute	Crowdedness			Camera Altitude			Times of the Day			Weather Condition				Occlusion		Scale Variation	
Sub-Attributes	1 Person	3 People	10 People	Low	Medium	High	Sunset/Sunrise	Midday	Night	Normal	Snow	Fog	Lightstorm	Low	High	No	Yes
# of Sequences	36	36	36	36	36	36	36	36	36	27	27	27	27	80	28	58	50

images purposed for person re-identification. Furthermore, a set of frame level annotations most of which identify miscellaneous environment parameters that were used to generate the frame are also included in the textual annotations of that frame. The frame level annotations include the environment type, weather and time of day markers, and applied post-fx presets (if any).

3.2. Experimental Analysis

In this section, using visual tracking as a test bed, we demonstrate how the proposed framework can be used to create realistic-looking and diverse synthetic datasets with auto-generated ground truth annotations. In our analysis, we specifically carry out two different sets of experiments. First, we demonstrate how our framework can be used to generate synthetic sequences with various challenging scenarios to evaluate the limits of state-of-the-art trackers (Sec. 3.2..3). Second, we show how our synthetically generated sequences can be utilized for training to boost the performance of deep-learning based visual trackers (Sec. 3.2..4). Before the analysis, we first briefly review the existing datasets proposed for tracking (Sec. 3.2..1) and present the evaluation measures used in our experiments (Sec. 3.2..2).

3.2..1 Existing Tracking Datasets

Tracking humans in videos is one of the most important topics in computer vision, with applications ranging from video surveillance to activity analysis. However, the widely-used benchmark datasets such as OTB100 [3], VOT [40, 41] and TC128 [4], which are indeed proposed for evaluating generic object trackers, have relatively small number of instances containing humans as objects of interest. Some datasets provide tracking sequences under very specific conditions, e.g. UAV123 [42] that presents sequences for low altitude UAV cameras and NUS-PRO [2] that contains videos that are mostly recorded by moving cameras. There exists some datasets that are specifically built for evaluating human trackers, such as

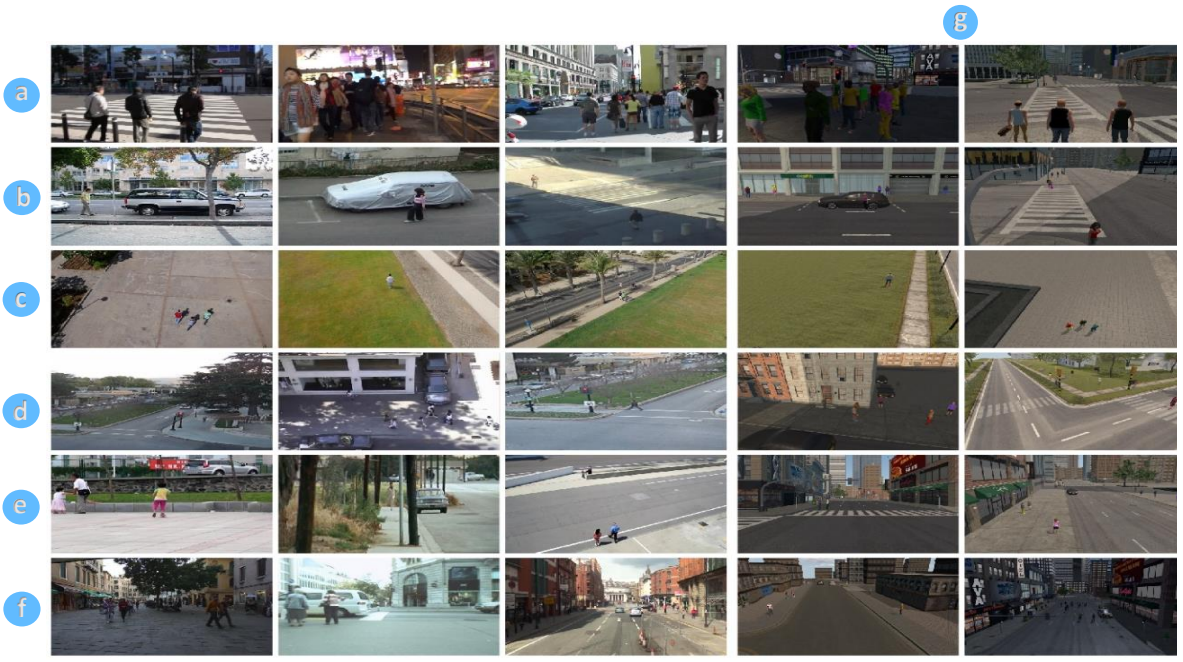


FIGURE 3.6.: Real vs. synthetic sequences. In terms of appearance, the sequences in (a) NUS-PRO, (b) TC128, (c) UAV123, (d) OTB100, (e) VOT, and (f) MOT datasets (first three frames in each row) are compatible with the synthetic ones produced by (g) NOVA (last two frames in each row).

DUKEMTMC [43], CamNeT [44], MOT [1] and NLPR-MCT [45], but these are mainly limited in both size and variability since obtaining annotated data for this task is difficult and time consuming. Either the sequences are captured with fixed cameras so the backgrounds are in general static or the lightning conditions do not vary much. To alleviate such shortcomings, in our experiments, we specifically focus on the task of tracking humans and use NOVA to generate two different datasets containing sequences with different levels of difficulty. Fig. 3.6. shows some sample sequences from our synthetic datasets, together with real-world sequences from NUS-PRO [2], TC128 [4], UAV123 [42], OTB100 [3], VOT [40, 41], and MOT [1] datasets. It is seen that NOVA is able to generate sequences that are compatible with the real-world sequences. We provide a more detailed comparison between our synthetic sequences and the curated real sequences used in the experiments in the supplementary material.

3.2..2 Evaluation Measures

In our experiments, we consider *precision* and *expected average overlap* (EAO), two commonly used metrics in evaluating visual trackers. Precision calculates the distance between the center of tracker bounding box and ground truth bounding box and checks whether this center error is within specified limits. We employ the conventional threshold of 20 pixels and consider the tracking as accurate for a frame if the center error is smaller than this value. We then extract the percentage of accurately predicted bounding boxes for each sequence in our dataset. EAO, on the other hand, is used to express accuracy and robustness of the tracker performance with a single score. At the beginning, the tracker is initialized and allowed to track the target until the end of the sequence or failure. When the tracker fails, it is reinitialized again and this process is repeated a number of times (3 times in our case). The mean of the average overlaps between the predicted and the ground truth bounding boxes gives EAO.

3.2..3 Using Synthetic Data to Evaluate Visual Trackers

Data Generation. To assess the limits of current state-of-the-art trackers, we use NOVA to generate a new synthetic dataset called VirtualPTB1 (Virtual Person Tracking Benchmark #1), unique in terms of its characteristics. As can be seen in Table 3..2, it includes sequences with different adverse weather conditions, crowdedness levels, and challenging factors due to different times of day and camera altitudes. VirtualPTB1 consists of 108 sequences, which are on average 5 secs long and have more than 13K frames altogether, along with per-frame bounding boxes for the persons of interest. The sequences are annotated with a total of 17 attributes from 6 different classes. Fig. 3.7. presents sample frames from VirtualPTB1 exhibiting the diversity and the photorealism of the generated sequences.

Visual Trackers. To analyze how the state-of-the-art generic object trackers perform on VirtualPTB1, we have selected six different correlation filter based tracking approaches, which perform well on the existing tracking benchmark datasets. These are ECO [46], $BACF$ [47], and context aware (CA) [48] versions of $MOSSE$, [49], DCF [50], $SAMF$ [51] and $STAPLE$ [52].

Results. In Fig. 3.8. and Fig. 3.9., we demonstrate the overall performances of the trackers on VirtualPTB1. As can be seen from Fig. 3.8., there are only a few sequences where the

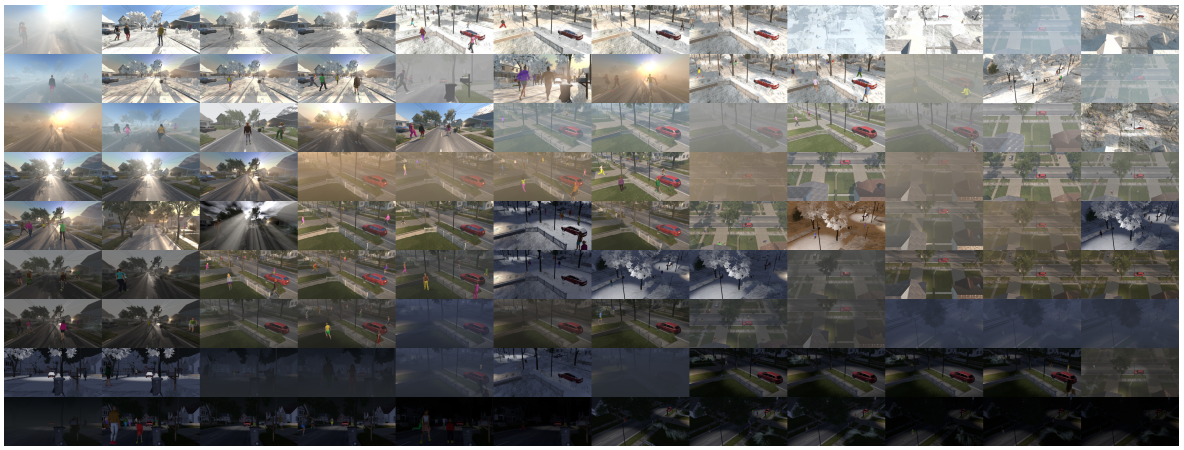


FIGURE 3.7.: VirtualPTB1, our proposed synthetic tracking dataset, consists of 108 sequences, each with a unique set of attributes. The first frames of each sequence are shown here, illustrating the variations in crowdedness, camera altitude, weather conditions and times of day.

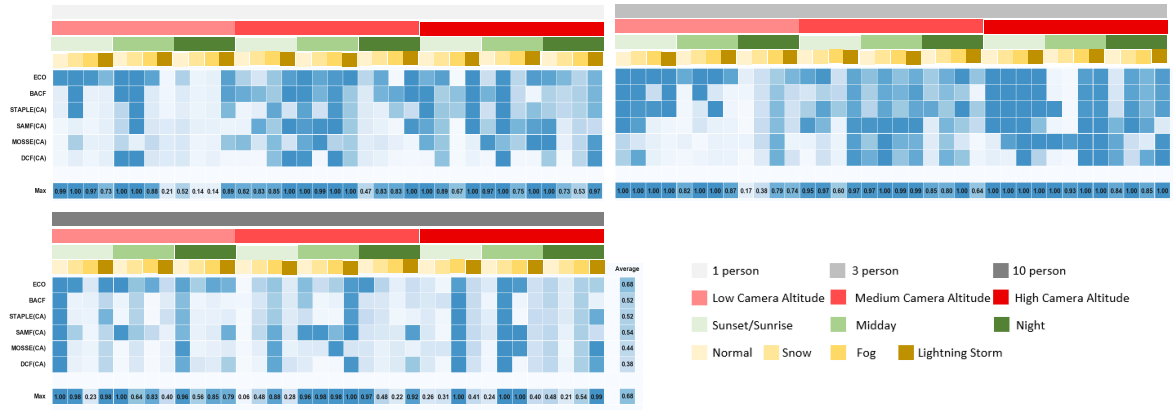


FIGURE 3.8.: Heatmap showing the precision of each tracker on each sequence of VirtualPTB1. The last row (Max) indicates the maximum performance achieved by the set of trackers on each sequence. The last column (Average) shows the average precision of a specific tracker over all sequences. Each color indicates different scene attribute. Gray, red, green and orange bars demonstrate scene crowdedness, camera altitude, time of day and weather condition, respectively, for a specific sequence below them by color variations that indicate their sub-attributes as given in the legend.

trackers give highly accurate results. In the remaining ones, they fail to precisely track the persons of interest, demonstrating how challenging VirtualPTB1 is. According to the precision rates, ECO tracker outperforms the others. BACF tracker and context aware versions of STAPLE and SAMF have nearly the same average precision scores although the sequences they show good performances are different. The examined trackers make use of different

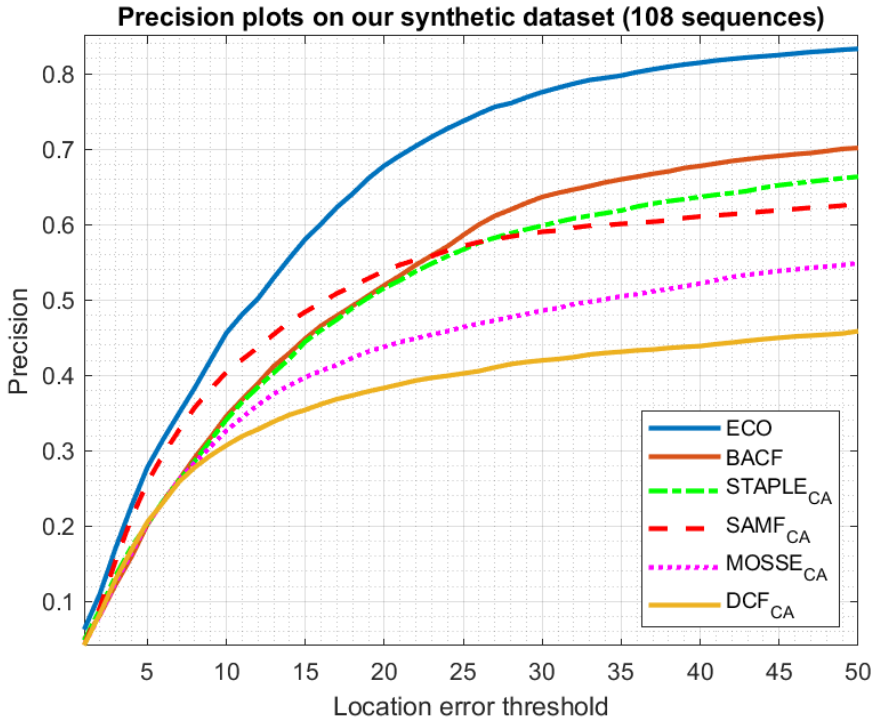


FIGURE 3.9.: Precision plot of the evaluated trackers on our dataset.

approaches and, hence, exhibit nonidentical performances on VirtualPTB1. Another key observation is that these scores are relatively low as compared to those reported in benchmark datasets containing real-world sequences [46–48]. This is in line with our design objectives for VirtualPTB1 as it introduces certain challenges which are mostly not present in the available benchmark sets. Sample qualitative tracking results can be found in the supplementary video.

Our detailed analysis reveals that tracking people in highly crowded scenes causes the trackers to lose the target very frequently as the persons of interest are highly likely to be occluded by the other persons. Moreover, it is noticed that the trackers perform poorly at night time and in foggy weather conditions. Under these circumstances, the trackers mostly cannot distinguish the tracked person from the background. Similarly, high camera altitude poses certain challenges as well since such altitudes cause the target to appear very small and, consequently, very hard to track. In Fig. 3.10., the corresponding precision plots for these challenging attributes are shown. Please refer to the supplementary material for an extended presentation and discussion of the results.

3.2..4 Using Synthetic Data to Train Visual Trackers

Data Generation and Collection. For our second set of experiments, we employed NOVA to generate a set of synthetic sequences that can be used to train deep learning based trackers. Here, we consider different training scenarios including synthetic and real sequences, and also a hybrid of those. In contrast to the former part, we carry our analysis on real test sequences for this set of experiments. In particular, NOVA is used generate 97 synthetic sequences and their ground truths annotations with pixel-level accuracy. However, to match the characteristics of the available real datasets, we limit the weather attribute to normal weather conditions, namely, clear-sky and three different variations of cloudy weather conditions. At the same time, we vary all other procedural generation parameters such as time of day, camera type, scene crowdedness and environment. In creating this set, it was aimed to mimic the general pattern of the existing real-world datasets, maintaining both the photorealism and the diversity at compatible levels.

In addition to the created synthetic dataset, we collect 125 real-world sequences from OTB100 [3], VOT [40, 41], TC128 [4], UAV123 [42], NUS-PRO [2] and MOT [1] datasets. We especially pick the sequences containing humans in outdoor environments and under normal weather conditions. Finally, we randomly divide these 125 real sequences into training and testing parts, where 97 sequences were selected for training and 28 for testing.

Please refer to the supplementary material for some sample frames from the synthetic and real-world sequences used. The synthetic sequences along with a file containing the links to the real-world sequences are provided at our project website under the name HybridPTB (Hybrid Person Tracking Benchmark).

Visual Trackers. We employ two state-of-the-art deep trackers in our experiments, namely CFNet [53] and DiMP [54]. Correlation filter based tracking (CFNet) is a deterministic, end-to-end representation learning tracker which considers correlation filter (CF) as a differentiable layer in a CNN architecture. This allows the error gradients to pass through the CF layer and tune the CNN features. DiMP, on the other hand, is a deep-learning based tracker that depends on Siamese architecture which accounts for the target and the background information while predicting the target object's location. The parameters of the tracker is learned in an end-to-end manner using a discriminative loss function.

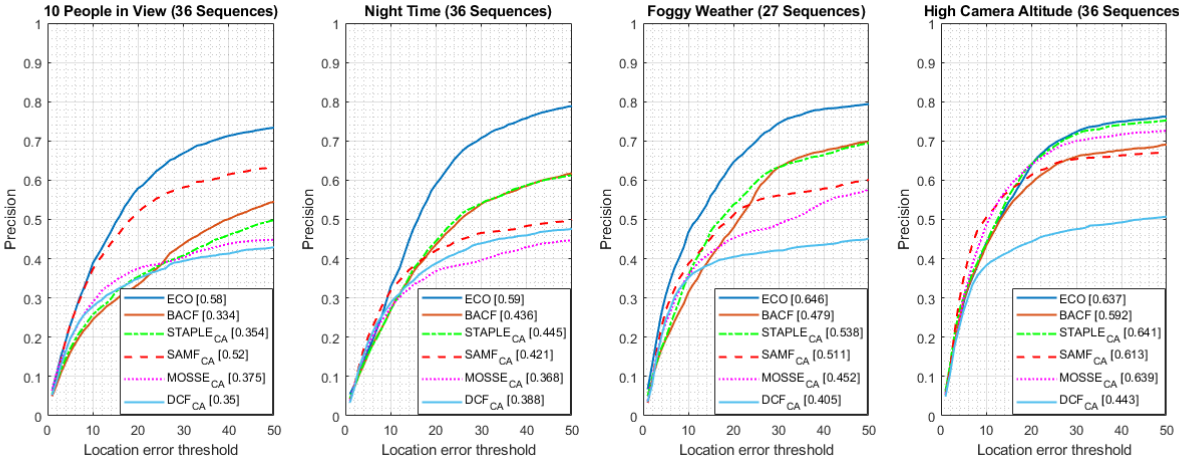


FIGURE 3.10.: Precision plots for the four challenging cases. Crowded scenes, night time, foggy weather and high camera altitude all cause a clear performance degradation.

Training Protocol. We consider training scenarios for the two deep trackers in two different schemes, as follows.

Training from Scratch. In the first scheme, we train each tracker from scratch by randomly initializing the model parameters using a different training set in each training scenario. The first scenario involves training the trackers using only the synthetic sequences generated by NOVA (E1). For the second one, the trackers are trained by employing only the real sequences from the training split of the dataset we collected (E2). Finally, in the last scenario, we consider a hybrid approach and explore the advantages of expanding the set of real sequences with the synthetic ones and training the trackers using this combined set (E3).

Fine-Tuning. For this scheme, instead of training the trackers from scratch, we perform fine-tuning considering their pre-trained versions again in three different scenarios. In the first and the second scenarios, the trackers are fine-tuned considering only the synthetic sequences (E4) and only the real training sequences (E5), respectively. The third scenario involves fine-tuning using the hybrid set containing both the synthetic and real sequences (E6).

Results. In Fig. 3.11., the results of our quantitative analysis are presented with the average overlap scores for DiMP and CFNet trackers obtained with each training scenario and compared to the baseline scores. Given the stochastic nature of DiMP tracker, we report the average and the standard deviation of its results for five repetitions. While training the trackers from scratch, using the synthetic sequences achieves better results as compared to using

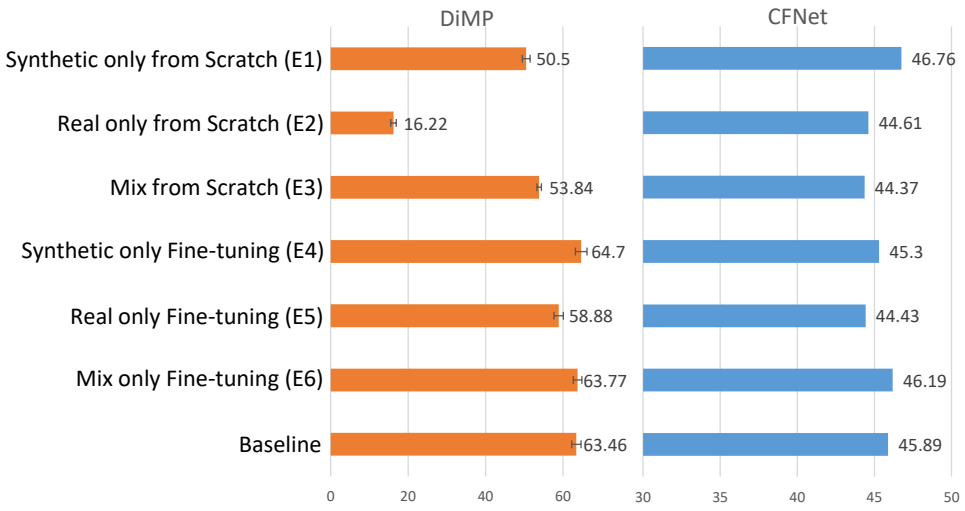


FIGURE 3.11.: EAO scores obtained with the six different training scenarios as compared to those of the baselines. Error bars on the DiMP results give the standard deviation of the EAO score. Fine-tuning the baselines on a mixture of synthetic and real sequences improves the performance. At the same time, training on synthetic sequences alone achieves better results compared to training solely on real sequences.

real sequences. Basically, this advantage can be attributed to the diverse and realistic nature of our synthetically generated sequences, which cover different environments, including indoor and outdoor ones, diverse weather conditions, multiple time of days, various camera types and distinctive humans. These factors enrich the generalization capability of the trained trackers, allowing them to learn better features and lead to more accurate results even on the real testing sequences. Moreover, comparable performances with the baseline models are achieved using only 97 synthetic sequences. Note that, in their original setting, the baseline CFNet model was trained using 3862 sequences with more than 1 million frames while the baseline DiMP model was trained by four different datasets, namely, LaSOT [55], GOT10k [56], TrackingNet [57], and COCO [10], which amount to a much larger set than the number of our training sequences. As for our fine-tuning experiments, we found out that fine-tuning the baseline models of DiMP and CFNet trackers on the mixture of synthetic and real sequences improves their performances to a greater extent as expected. The gain is especially significant for CFNet, whose baseline model was pre-trained on ILSVRC Video dataset that does not contain humans as objects of interest. Another important observation is that fine-tuning the baselines only on our synthetic sequences seems more advantageous than fine-tuning on real-world sequences alone. This further demonstrates the advantage of using our synthetic data. It is worth noting that, these results are also taken to indicate that the domain gap due to the differences between the synthetic and the real-world sequences

seems to be minimal. Although the trackers were trained on NOVA's synthetic sequences and testing was carried out on real-world sequences, *i.e.*, our training and test sequences do not share the same level of photorealism, it is seen that using synthetic person sequences during training let the trackers learn more fine-grained features for person tracking, and, in return, leads to better performances.

3.3. Discussion

As a case study, we considered visual tracking and employed our proposed NOVA framework to create two different datasets for different purposes. The first dataset, VirtualPTB1, includes 108 sequences with automatically generated ground truths and and a total of 17 scene level attributes. Under short-term tracking scenarios, the sequences demonstrate a wide variety of factors including weather conditions, times of day, overall crowdedness of the scene, camera altitude, occlusion and scale variation. Our thorough analysis of various state-of-the-art trackers on VirtualPTB1 sheds light on trackers' weaknesses in adverse conditions such as high crowdedness, high camera altitude, night time, and foggy weather. Our second synthetic dataset, on the other hand, consists of 97 sequences with normal weather conditions. We have used this dataset to train two deep trackers, CFNet and DiMP. Our results reveal that using our synthetic sequences during training leads to a performance boost in several aspects for both of these trackers. Thus, it is shown that the variety and the level of realism of the scene attributes in our dataset make it a good proxy of the real-world for evaluating and training visual trackers.

3.4. Conclusion

In this work, we have presented a novel engine called NOVA for creating photorealistic 3D rendered worlds containing synthetic humans, along with ground truth annotations at scene, object and pixel -levels. The proposed framework automates data collection and labeling pipeline for a wide range of low and high-level computer vision tasks. In particular, the engine emphasizes procedural generation of humans, which makes NOVA unique compared to existing systems. It allows to produce diverse arrays of human agents, in terms of body shape, clothing, gender and age characteristics, accessories and action variety. Moreover,

NOVA allows to play with weather and illumination conditions within the created 3D virtual worlds, establishing it as a test bed for evaluating adverse cases such as low light, nighttime, rain, snow, or fog. These capabilities make NOVA a distinct and versatile framework to quickly generate arbitrarily large amounts of synthetic data for a multitude of computer vision tasks. These large synthetic datasets can be used in model training to boost the performance of state-of-the-art learning based computer vision models. Our results show that the scenes that are either highly crowded, or taking place at night or at foggy weather conditions pose certain challenges for the state-of-the-art trackers. It is also seen that using synthetic data generated by NOVA for training can boost the performance of learning-based trackers on real videos.

An online demo of NOVA and videos illustrating NOVA's capabilities are available at the project website `https://graphics.cs.hacettepe.edu.tr/NOVA` along with VirtualPTB1 and HybridPTB, featuring the synthetic sequences generated by NOVA for the first and the second set of experiments, respectively.

Chapter 4.

NOVA Adverse Weather Conditions

FIGURE 4.1.: On the left half, sample frames from the currently-available real (top-left quarter) [1–4] and synthetic (bottom-left quarter) [5–8] visual object tracking datasets demonstrate the lack of adverse weather conditions. The right half presents sample frames from sequences spanning raining, foggy and snowy weather conditions from PTAW172Real (top-right quarter) and PTAW217Synth (bottom-right quarter) datasets that we introduce in this work.

4.1. Extensions to NOVA Framework

To procedurally generate synthetic sequences of pedestrians under adverse weather conditions, we use the NOVA rendering engine [58], which is designed with the goal of allowing researchers with no experience in computer graphics to generate high quality datasets with accurate and dense annotations. NOVA operates in two modes. The first is to generate a single sequence while the other is to generate a full dataset. The first mode gives the user full control of the sequence being generated where it is possible to specify the environment, the weather condition, time of day, camera type, number of cars and number of pedestrians and their density. The dataset mode requires nothing to be specified except the number of sequences to be generated so that NOVA varies the other parameters automatically.

For the particular task of person tracking this work deals with, NOVA generates, for each frame, a bounding box specifying the exact location of the person(s) being tracked in the frame and the occlusion state, that is, whether any other object or person in the scene occludes the person(s) being tracked at that instant. In addition to these, a supplementary metadata are provided with each sequence denoting the environment, weather condition, time of day, camera type, number of people and cars and people density.

One of the major highlights of NOVA is its capacity to procedurally generate highly diverse and photorealistic sets of synthetic humans. So much so that, each generated human is practically unique in appearance due to the practically infinite number of recipes (combinations of parameters that are assigned randomly on the fly but in cohesion with each other) that NOVA uses in creating them. In this work, we further develop this aspect of NOVA by incorporating premade synthetic humans from Microsoft Rocketbox Avatar Library [59].

Since the main aim of this work is to enhance the performance of the trackers under adverse weather conditions, we also extended other capabilities of NOVA toward photorealistic simulation of the generated humans under adverse weather conditions. The environment is built to change dynamically to match the corresponding weather condition and time of the day. Accordingly, the textures of buildings are changed to have lit windows at nighttime. Furthermore, we implemented the following for the three weather conditions to facilitate the generation of synthetic sequences with similar visual characteristics to the ones observed in the real-world videos captured under adverse weather conditions.

Snowy Weather Condition. First, the variety of clothing used to generate humans in snowy weather is restricted only to outdoor cold-weather clothes. At the same time, humans are randomly assigned umbrellas. An umbrella is attached to the right or left hand at random. The animation of the character is set to match the umbrella mode, *i.e.*, open or close. Snow tracks left by cars and pedestrians are simulated. Furthermore, snow banks and melt snow are created on the pavements and roads to give a higher degree of realism. For this, a set of street light poles in the scene are selected at random to determine the positions of the snow banks. Then, from a predefined set of snow banks, one snow bank is instantiated for each position. After that, snow materials are assigned at random to the snow banks. Following this, the scale and rotation of these models are randomized to allow for even more diversity. On the other hand, the melt snow is simulated by the same snow shader that is used to simulate accumulated snow but with the accumulation parameter set to a random smaller number than the one used for accumulated snow. Making use of the particle system and post-processing effects, falling snow particles and blizzard were randomly introduced to the simulation, as well.

Rainy Weather Condition. Similar to the snowy weather condition, humans in rainy weather are also generated with outdoor cold-weather clothes; and umbrellas are given to some of the generated humans in the same way. In addition, water puddles are simulated to account for water accumulation due to the rain. This is realized by using a puddle shader that is assigned to some of the ground materials (pavements, roads etc.) randomly. For the heavy rain, the rain splash is activated and additional water puddles are instantiated from a predefined set of water puddles. Rain drops are generated using the particle system. Furthermore, rain drops falling on camera lens are simulated using post-processing effects to match the characteristic of the rainy videos in real life.

Foggy Weather Condition. The clothes of the people produced in the foggy weather simulation are not limited to a specific category, but instead are randomly selected. Additionally, the fog is simulated using post-processing effects and the Enviro system [36]. The fog density is randomized at run time to give more diversity.

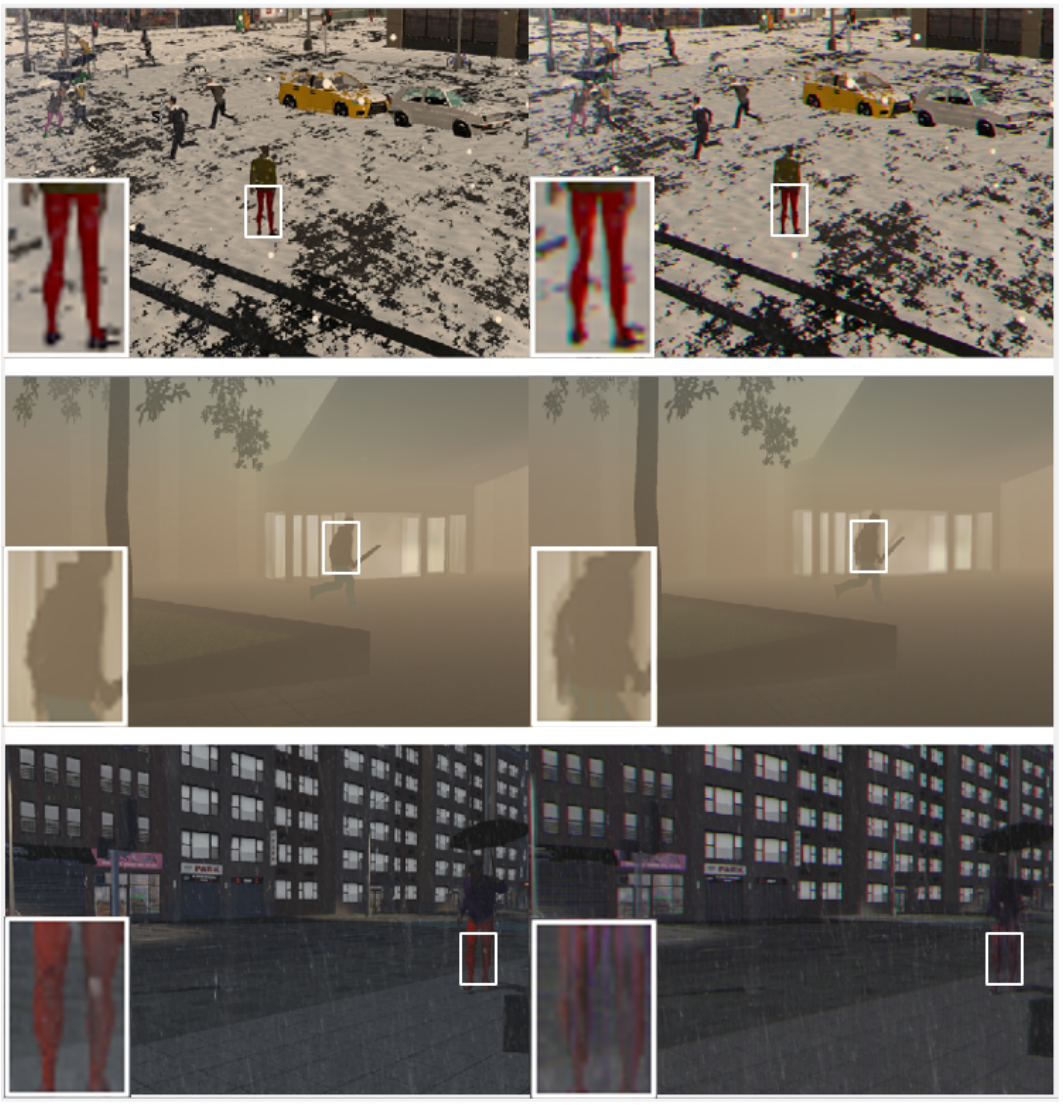


FIGURE 4.2.: Chromatic aberration, motion blur and both effects are demonstrated in the first, second and third rows, respectively. The first column shows the original frame while the second displays the result of applying the effect(s).

Motion Blur and Chromatic Aberration. These camera effects were simulated additionally to match the camera degradation observed in real-life adverse weather videos. Using post-processing, NOVA simulates these two effects procedurally and parametrically. Thus, how severe the effect of these two degradations is randomly configured at run time to provide further diversity in the generated synthetic sequences. In Fig. 4.2., the impact of using these effects over the generated sequences is shown with a sample of images.

TABLE 4..1: Dataset statistics of PTAW172Real.

Class	Min Frames	Max Frames	Mean Frames	Total Frames	Videos
Rain	108	1755	498	31888	64
Snow	113	960	394	24010	61
Fog	106	750	328	15394	47
All	106	1755	407	71292	172

4.2. PTAW172Real and PTAW217Synth Datasets

4.2..1 Real-World Data Collection for PTAW172Real

For the aim of analysing the performance of the recent general purpose visual trackers under adverse weather conditions, we collected real-world videos from YouTube spanning snowy, rainy and foggy weather. Keywords such as *"adverse"*, *"extreme"*, *"heavy"*, and *"severe"* were used together with the weather names to initiate searches on the Youtube video-sharing platform. Following this, the query results were checked and only the videos satisfying the adverse weather conditions were selected. The acquired videos were edited to assure that the object is not occluded and clearly visible in the initial frame. At the same time, the lengths of the videos were modified as needed to keep them around 400 frames per video to provide compatibility with the sequences in the available visual object tracking datasets. Statistics showing the minimum, maximum, average and total number of frames are given in Table 4..1. The number of videos in the dataset is 172 and the total number of frames is over 71 thousand. The collected videos are at 24 frames per second (FPS) and average time period per sequence is around 17 seconds. Sample frames from the collected PTAW172Real dataset are shown in Fig. 4.3..

We used the VGG Image Annotator tool [60, 61] for annotating the dataset. We annotated every 5th frame be drawing a bounding box around the person of interest. The accessories such as handbag etc. that a person can carry were excluded and the tightest box was drawn. When the person was partially or fully occluded, the estimated location of the person was considered. Additionally, each video was associated with four attributes regarding object occlusion, scale change, background clutter and abrupt camera motion. Fig. 4.4. gives the hierarchical distribution of the attributes in PTAW172Real dataset.

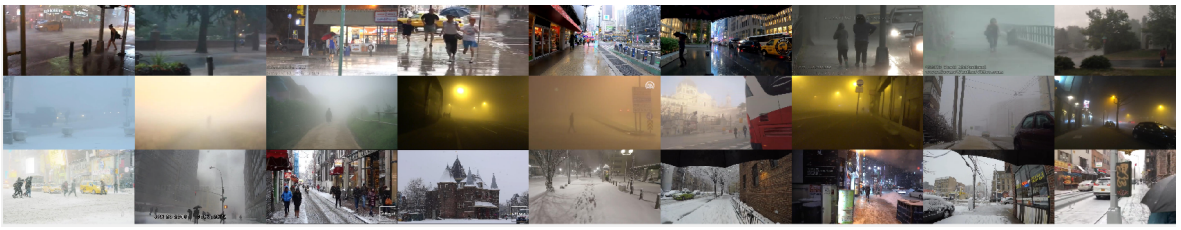


FIGURE 4.3.: PTAW172Real, our real person tracking dataset, consists of 172 sequences. Each row shows a specific adverse weather condition, namely rain, fog, and snow.

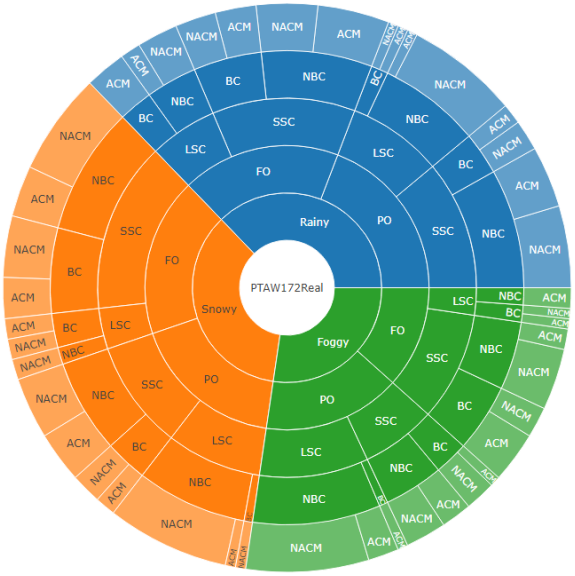


FIGURE 4.4.: The sunburst chart shows the different attributes distribution across PTAW172Real dataset. The inner circle shows the weather conditions, outer circles show occlusion (FO:Full Occlusion, PO: Partial Occlusion), scale change (LSC: Large Scale Change, SSC: Small Scale Change), background clutter (BC: Background Clutter, NBC: No Background Clutter) and abrupt camera motion (ACM: Abrupt Camera Motion, NACM: No Abrupt Camera Motion).

4.2..2 Synthetic Data Generation for PTAW217Synth

PTAW217Synth employed in the experiments to train the deep learning trackers consists of 217 synthetic sequences that were generated using the NOVA rendering engine. NOVA allows to specify the attributes of the sequences to be generated. In this work, we configured these attributes to match our goal of generating diverse synthetic sequences under adverse weather conditions. Accordingly, the weather conditions were limited to snowy, rainy and foggy weather. The virtual camera type to capture the simulations was set as either the street-level camera or the surveillance camera. The simulation environment was limited to

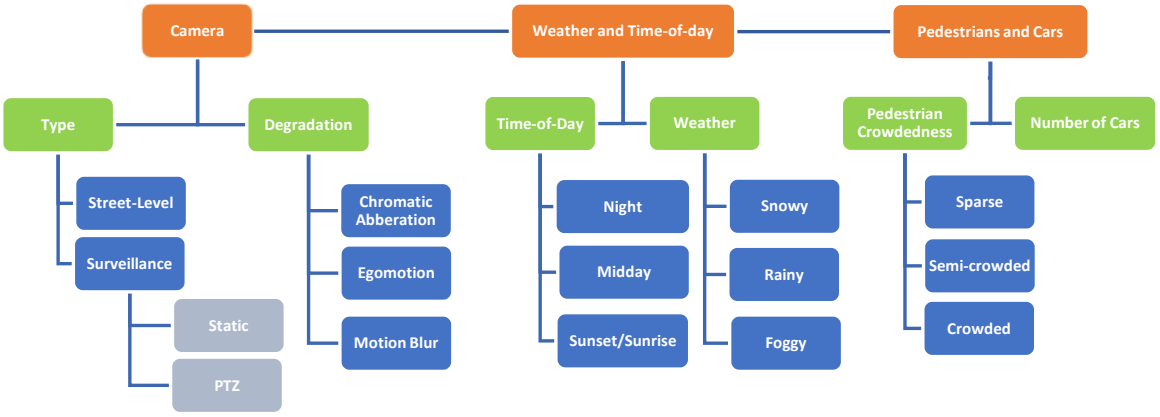


FIGURE 4.5.: Hierarchical view of the attributes across our training synthetic person tracking dataset, PTAW217Synth, generated by using NOVA.

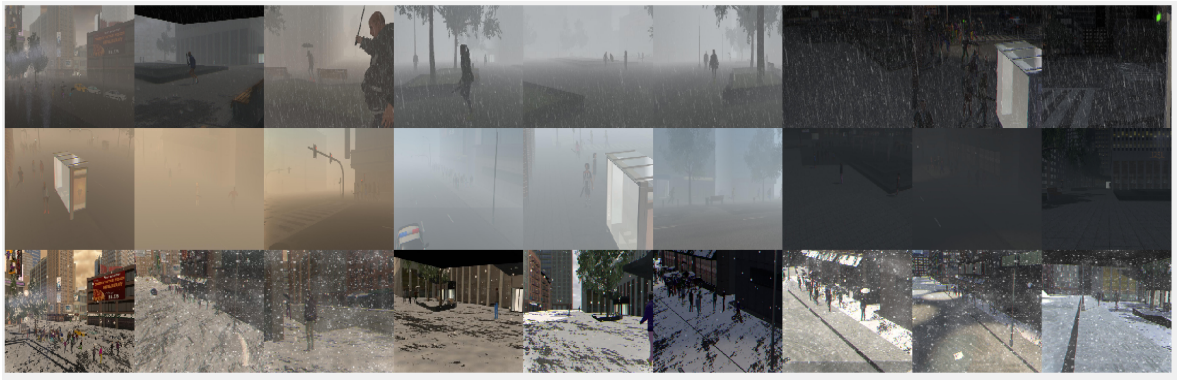


FIGURE 4.6.: PTAW217Synth, our training synthetic tracking dataset, consists of 217 sequences, each with a unique set of attributes. Random frames are shown here, illustrating the variations in crowdedness, camera altitude, weather conditions and times of day.

the streets of an urban center, since such are the most common settings in the real-world visual object tracking datasets. In parallel to this, all other attributes such as time of day and crowdedness were randomised to ensure the diversity of the generated sequences. The attributes of the generated synthetic sequences are given in Fig. 4.5.. Consequently, the diversity of the generated sequences can be noted in the sample images from these sequences in Fig. 4.6..

Further information regarding the minimum, maximum, average and total number of frames are shown in Table 4..2. The overall average number of frames per sequences is 500 which translates to a duration of 21 seconds as the sequences were generated at 24 FPS. The total number of frames of the 217 sequences within the dataset is more than 108 thousand.

TABLE 4..2: Dataset statistics of PTAW217Synth.

Class	Min Frames	Max Frames	Mean Frames	Total Frames	Videos
Rain	490	510	501	34538	69
Snow	490	510	501	37577	75
Fog	490	510	499	36432	73
All	490	510	500	108547	217

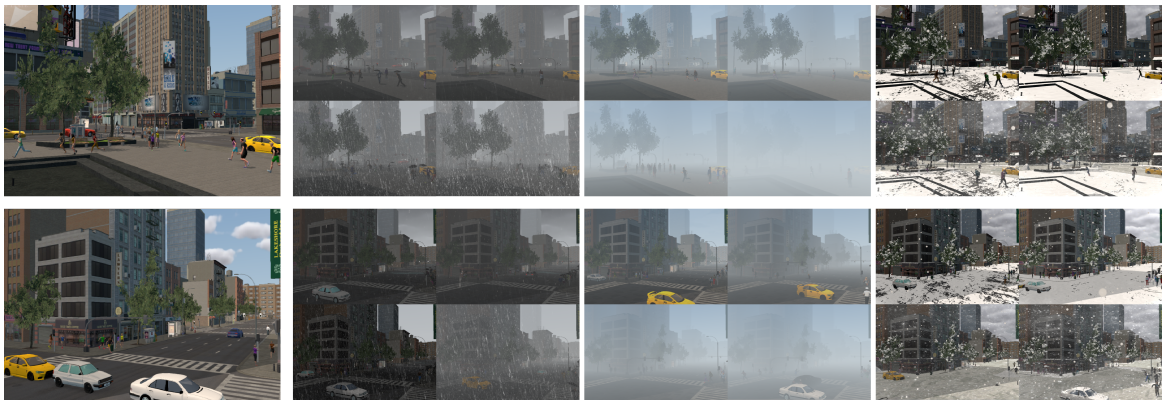


FIGURE 4.7.: The figure demonstrates the weather variations simulated in PTAW217Synth. The first and second rows present different view points of the same location. Each group of 2x2 images shows one weather condition (from left to right: rainy, foggy, and snowy) in increasing adversity while the leftmost image shows the same location in clear weather.

We should note that PTAW217Synth has an even distribution of sequences across the rainy, snowy and foggy weather conditions. The sample images captured at a single location from two different view points given in Fig. 4.7. further demonstrate the variety of the simulated weather conditions.

A visual comparison between PTAW172Real and PTAW217Synth datasets is given in Fig. 4.8.. In each row a specific weather condition is presented. Both datasets exhibit similar visual characteristics for the three weather conditions. The figure also demonstrates the level of photorealism of the PTAW217Synth dataset.

4.3. Experiments

In this section, we study the performance of the state-of-the-art visual trackers in adverse weather conditions. The poor performance is highlighted and discussed. In the second set of

FIGURE 4.8.: A visual comparison among the synthetic PTAW217Synth (to the right) and real PTAW172Real (to the left) datasets. Each row demonstrate specific weather condition (from top to bottom: rainy, foggy, and snowy).

experiments, we show how the performance of the deep-learning based visual trackers can be enhanced by training on our generated synthetic sequences. First, the evaluation measures are discussed in Section 4.3..1. Then, the utilized trackers are described in Section 4.3..2 and the training protocol is explained in Section 4.3..3. Finally, the results are analysed and explored in Section 4.3..4.

4.3..1 Evaluation Measures

The two widely used metrics *precision* and *success* (IoU) are employed for evaluating the performance of the visual trackers analyzed in this work. Precision calculates the distance between the centers of the tracker bounding box and the ground truth bounding box and then checks whether this center error is within the specified limits. We employ the conventional threshold of 20 pixels and consider the tracking as accurate for a frame if the center error is smaller than this value. We then extract the percentage of the accurately predicted bounding boxes for each sequence in our dataset. On the other hand, success measures the intersection over union (IoU) of the tracker and ground truth bounding boxes. We take a tracking to be successful if the IoU is larger than the common threshold of 0.50, and report the percentage of the successfully predicted bounding boxes averaged over the sequences in our dataset.

4.3..2 Trackers

In order to properly address the poor performance of the state-of-the-art general purpose trackers under adverse weather conditions, two different sets of trackers were selected. The sets present the two main approaches in visual object tracking, *i.e.* correlation filter -based and learning -based tracking.

Five different state-of-the-art correlation filter based trackers were chosen for the experiments. These are ECO [46], $BACF$ [47], and context aware (CA) [48] versions of DCF [50], $SAMF$ [51] and $STAPLE$ [52]. DCF, dual correlation filter, utilizes a kernelized correlation filter (KCF) that has a similar complexity to the linear counterpart of it, which improves tracker speed (FPS) considerably. On the other hand, $SAMF$, scale adaptive with multiple features, uses a scale adaptive template size instead of using a fixed one for the correlation filter kernel which is stated to make the tracker more robust. $STAPLE$, sum of template and pixel-wise learners, fuses template and histogram scores to better handle shape deformation which facilitates tracking deformable objects more accurately. ECO uses a modified version of DCF to improve memory usage, tracking speed, and robustness. $BACF$ uses a background-aware correlation filter that utilizes specific manually extracted features that account for both background and object of interest change over time. The context aware versions of DCF [50], $SAMF$ [51] and $STAPLE$ [52] that we used improve the original implementations by utilizing the global context information into the standard correlation filter tracking algorithms.

Similarly, for investigating the benefits of training on our generated synthetic sequences, four state-of-the-art learning based deep trackers were used. They are DiMP [54], ATOM [62], PrDiMP [63], and KYS [64]. DiMP is an offline learning based tracker that can be trained in an end-to-end manner. It applies both background and target information in the process of predicting the object of interest location. The tracker is based on the Siamese tracking architecture. It learns the discriminative loss function during the training phase. ATOM, however, is a deep-learning tracker that is trained both offline and online. Its tracking algorithm deploys target estimation and classification that are learnt offline and online respectively. At run-time, the classification component predicts the IoU between the target object and the estimated bounding box. PrDiMP is another learning based tracker that is based on the DiMP architecture. However, unlike DiMP tracker, PrDiMP applies probabilistic regression concept and predicts the probability density of the target given the input frame. This tracker

is trained by minimizing KL-divergence in offline manner. KYS tracker, however, uses the visual scene information to better enhance the target localization and tracking. KYS encodes this information using localized state vectors and propagates it through the sequence to achieve better knowledge of the scene. Thus, it achieves better performance during testing. KYS is trained offline to learn how to propagate the scene information.

4.3..3 Training Protocol

We perform two training scenarios to assess the benefits of the generated synthetic sequences when used for training visual object trackers. For both experiments, the training was done using the whole PTAW217Synth dataset of 217 synthetic sequences. At the same time, the validation and testing were performed on the whole PTAW172Real dataset. For validation, 33 videos spanning the rainy, foggy and snowy weather conditions were selected at random. While the remaining 139 videos were applied for testing.

Training from Scratch. In the first scenario, we train the trackers from scratch using only the generated synthetic sequences. Then, the best model on the validation set is tested on the test set. The mean and the standard deviation of the tracker performances are reported for 5 iterations to account for the stochastic nature of these trackers. Both validation and test sets are real and contain no overlapping videos.

Fine-Tuning. In the second scenario, the pre-trained versions provided by the authors of the four trackers are fine-tuned on our synthetic sequences. Later, the performance of these models are stated as done in the previous case.

4.3..4 Results

The performance in terms of precision and success score are shown in Tables 4..3 and 4..4 for the studied trackers on the test partition of PTAW172Real, namely 163 videos. These results show that the trackers from both tracking mainstreams, correlation filter based and learning based, performed poorly under adverse weather conditions. This observation confirms that adverse weather conditions pose certain challenges for the state-of-the-art tracking

TABLE 4..3: Precision results of the available state-of-the-art trackers on the adverse weather condition real dataset, test partition of PTAW172Real.

Class	ECO	BACF	STAPLE_CA	SAMF_CA	DCF_CA	ATOM	DiMP	PrDiMP	KYS
Rain	0.59	0.50	0.46	0.38	0.22	0.61+/-0.01	0.60+/-0.01	0.61+/-0.01	0.63+/-0.02
Snow	0.56	0.53	0.49	0.46	0.35	0.60+/-0.01	0.62+/-0.01	0.59+/-0.01	0.58+/-0.01
Fog	0.67	0.65	0.59	0.42	0.37	0.73+/-0.01	0.74+/-0.01	0.74+/-0.01	0.77+/-0.02

TABLE 4..4: Success scores of the available state-of-the-art trackers on the real adverse weather condition dataset, PTAW172Real.

Class	ECO	BACF	STAPLE_CA	SAMF_CA	DCF_CA	ATOM	DiMP	PrDiMP	KYS
Rain	0.64	0.56	0.47	0.45	0.20	0.66+/-0.01	0.63+/-0.01	0.64+/-0.01	0.65+/-0.02
Snow	0.56	0.55	0.49	0.43	0.28	0.59+/-0.01	0.61+/-0.01	0.59+/-0.01	0.57+/-0.01
Fog	0.70	0.69	0.59	0.42	0.27	0.73+/-0.01	0.73+/-0.01	0.73+/-0.01	0.78+/-0.02

algorithms. The correlation filter trackers perform worse than the deep trackers because they are mostly online learning trackers. On the other hand, the deep trackers, which are based on offline learning algorithms, were trained on large scale datasets, which may have contained a number of videos under adverse weather conditions. Thus, they performed slightly better than the correlation ones.

It seems that rain and snow particles, that partially occlude the object of interest, cause a significant change on the visual characteristics of the trackers. Thus, it makes it hard for the tracker to differentiate the target object from the background. This effect is particularly clear when the size of the object of interest is relatively small. In parallel to that, fog causes both the background and the object of interest regions to have similar visual appearance. Thus, it makes it hard for the tracker to distinguish the target object from the background. Even so, foggy weather condition seems to be slightly less challenging as compared to the others.

The results of our training experiments are shown in Fig. 4.9.. The IoU scores for the four trained trackers, namely DiMP, ATOM and PrDiMP, are presented for the two training scenarios. Moreover, these results are compared to the ones of their corresponding baselines. Both average and standard deviation on five iterations were reported to account for the stochastic nature of these trackers. Training these trackers from scratch on our adverse weather synthetic sequences achieves comparable results to the ones obtained using the baseline for DiMP and PrDiMP. For ATOM and KYS, however, the trained models from scratch surpassed their baselines. On the other hand, fine-tuning the pre-trained models on our synthetic sequences improved the performance of the three trackers ATOM, DiMP and PrDiMP distinctly.

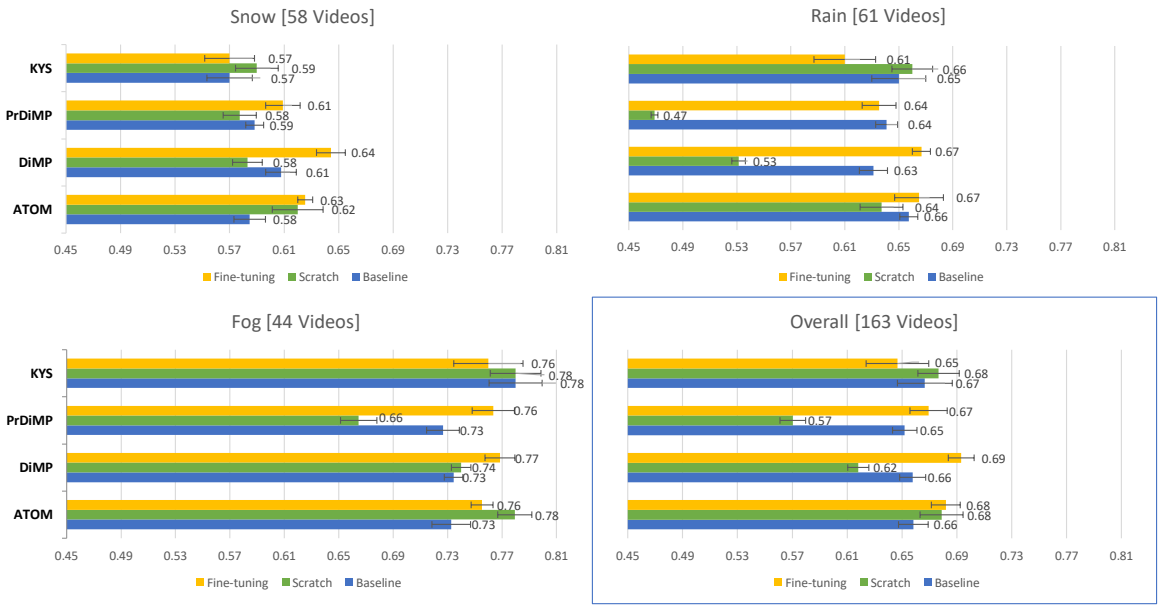


FIGURE 4.9.: IoU results obtained with the two different training scenarios as compared to those of the baselines. Error bars give the standard deviation of the IoU results. Fine tuning the baselines on our synthetic sequences improves the performance.

It is worth noting, that both the tracking algorithm and the training dataset affect how a specific tracker gains from training on our synthetic sequences. Both determine which training scenario, from scratch or fine-tuning, is more beneficial. For example, DiMP and PrDiMP trackers got the most advantage from fine-tuning. On the other hand, training from scratch was better for KYS tracker, while the performance of ATOM was improved in both scenarios. Another point to be noticed is the conspicuous difference in the level of improvement in trackers performance across different weather conditions. This can be directly linked to the varying distribution of the adverse weather conditions in the different training datasets used for these baselines. So much so that, the lack of adverse weather conditions videos in the training dataset stands out to be the main reason behind the observed performance boost since using even a relatively small number of synthetic sequences spanning these absent features helped the trackers to outperform their baselines, given that the trackers were originally trained on large scale datasets such as LaSOT [55], GOT10k [56], COCO [10], and TrackingNet [57], each far exceeding PTAW217Synth in number of sequences.

It is important to note that test set contains only real sequences. Thus, the domain gap problem is not a playing factor under the scope of this analysis. In contrast, diversity of the synthetic sequences in terms of weather conditions, times of day, lighting conditions, camera

attributes and synthetic humans altogether enhanced the training process significantly. Additionally, the high level of photorealism of these synthetic sequences mitigated the gap across the real and synthetic domains. Thus, training from scratch or fine-tuning on our synthetic sequences directly improved the trackers performance.

A qualitative comparison among the tracking results achieved by the baselines and the trained models is presented in Fig. 4.10.. It is seen that utilizing our synthetic data for training improves the performance of the baselines under adverse weather conditions.

Additionally, Fig. 4.11. displays the success scores for the four deep trackers under full occlusion, scale change, background clutter and sudden camera motion videos. In general, both the baselines and the trained models performed the worst in sequences with background clutter while the ones with sudden camera motion resulted in relatively higher performance. It could be because the background clutter under adverse weather conditions causes the trackers to experience a significant difficulty in locating the object of interest since both have similar visual appearances. On the other hand, the reason that abrupt camera motion does not seem to be effecting trackers as much as the other attributes could be due to the fact that the other three attributes are more closely associated with the object of interest as compared to the camera motion which effect both background and target similarly. A table showing the number of sequences in each weather condition for each of the four attributes is provided in the supplementary material.

4.4. Conclusion

Our work investigated the lack of adverse weather conditions in the available general purpose visual tracking datasets and highlighted the low performance of the state-of-art trackers under these specific circumstances. As a solution, we proposed using our NOVA rendering engine to generate synthetic sequences that span snowy, rainy and foggy weather conditions. We trained four different deep trackers, namely DiMP, ATOM, KYS and PrDiMP, on 217 synthetic sequences generated by NOVA and tested them on the real videos that were collected from YouTube and annotated meanly by us for that aim. Our analysis reveals that applying our synthetic sequences for training purposes can bridge the data gap and improve the trackers performance considerably.

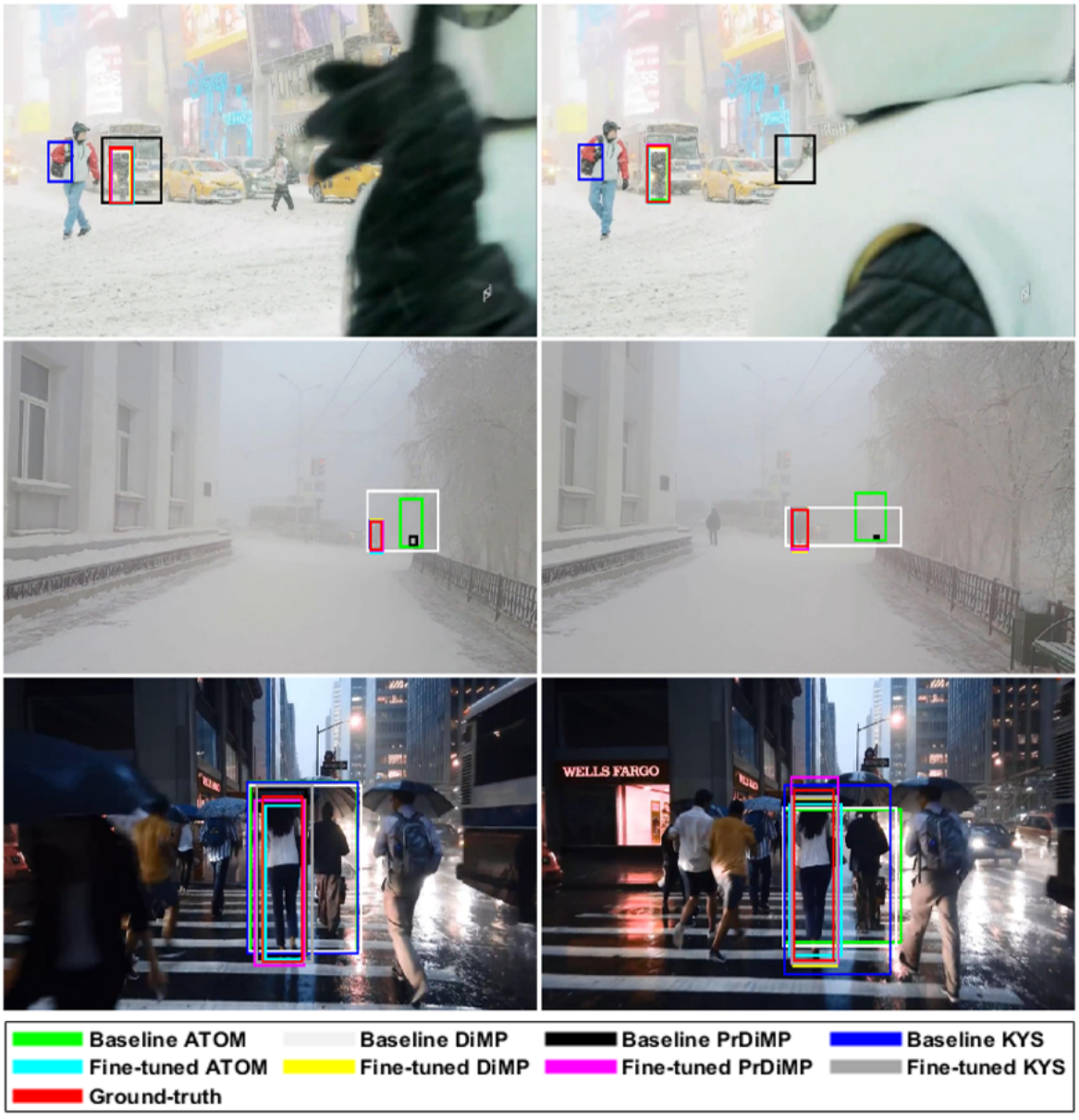


FIGURE 4.10.: A qualitative comparison of our trained trackers with the baselines on three example sequences. Training on PTAW217Synth improves the trackers performance under adverse weather conditions.

The datasets PTAW172Real and PTAW217Synth that we featured in this work are available for download at the project website `https://graphics.cs.hacettepe.edu.tr/NOVA-Adverse` along with a supporting video illustrating the motivation behind this work, a sample of sequences from PTAW217Synth and also a sample of the PTAW172Real sequences superimposed with tracking results.

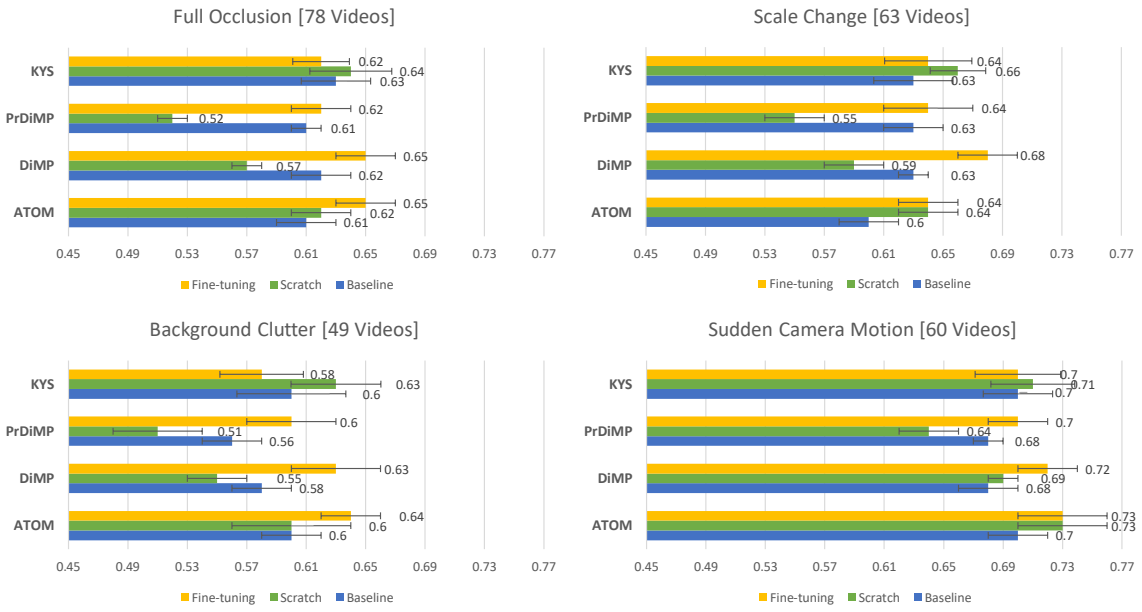


FIGURE 4.11.: Success scores for ATOM, DiMP, PrDiMP and KYS trackers are shown for four different attributes. Background clutter causes the trackers to perform poorly.

Chapter 5.

Limitations and Future Work

Investigating the usability of synthetic data generated by the NOVA rendering engine is an important aspect of this study. Here, we demonstrated that using synthetic data generated by NOVA can both boost the performance of the state-of-the-art trackers and provide a better medium for testing tracking algorithms under a number of challenging attributes. However, one concern is the generalizability of these findings to other computer vision tasks such as semantic segmentation, depth estimation and so on. Considering the procedural generation capabilities of NOVA, including rich variety of annotations it can produce, there are various other directions that can be explored to thoroughly address the matter. Accordingly, it is our plan to extend this study toward exploring other computer vision tasks in future works.

The lack of performance of the trackers on NOVA's synthetic test sequences could be partially attributed to the domain gap problem, as the trackers were trained on real-world data. However, the photorealism of the generated sequences is expected to have mitigated this gap. In parallel to that, the improvement in performance of the deep trackers on tests with real-world data upon having been trained on the synthetic data sheds light on the cohesion of the synthetic data with real-world data. In addition, the fact that not just the deep trackers but also the correlation-filter -based trackers, which rely solely on online learning, showed poor performance on NOVA's synthetic test sequences further signifies that the main factor at play is the challenging nature of these sequences as the domain gap is not thought to cause such a clear degradation in performance across the board.

Similar to the previous discussion, perhaps the domain gap problem in [65] is the one of central concern in this scope as well. It arises mainly because the training and testing processes

take place in two different domains i.e. synthetic and real domains, respectively. To address this point, we paid great attention to the photorealism of the generated synthetic sequences and most specifically the simulated adverse weather conditions. The second key issue is that synthetic sequences are usually generated at optimal lighting and recording conditions. Thus, the lack of image artifacts such as motion blur, chromatic aberration, noise and others may cause models trained on it to fail once such artifacts are encountered in real sequences. To mitigate this problem, we generate our synthetic sequences at different lighting conditions and recording setups. Additionally, we simulate lens artifacts such as motion blur and chromatic aberration. Another note-worthy issue is the fact that repetitive textures, objects, animations, and motions frequently observed in virtual 3D worlds may cause over-fitting. We tackled this issue by diversifying scene elements such as pedestrians, buildings, cars, and other scene objects.

Throughout this work, we demonstrated how our generated synthetic sequences improved trackers performance on adverse weather conditions. However, investigating the effect of adverse weather conditions on other computer vision tasks like optical flow estimation, depth estimation, and person re-identification are sill open questions. The boost in performance upon remedying the lack of sample with adverse weather conditions for the VOT task could be an indication of a similar problem in other computer vision tasks. For DiMP tracker,the performance of the tracker when trained from scratch on real data and its baseline both show clearly different results. This is expected since the baseline was trained on different datasets as compared to the ones used in E2 experiments i.e. training on real data. It is an interesting point to see that training on synthetic sequence achieves better results as compared to training on the same number of real videos. That was linked to the fact that the real visual object tracking datasets focus more on the standard and normal tracking scenarios. However, with our synthetic data, more diverse attributes were attained such as different illumination and weather conditions, various crowdedness levels, and camera setups. It is also interesting to explore whether using our synthetic data for augmenting the lack of enough samples in one class (cars) could improve the performance of computer vision algorithms on other classes (pedestrians).

In the light of this study, we believe that using our rendering engine NOVA to generate synthetic training data can bridge the gap of data scarcity in said tasks toward improvement in both accuracy and robustness.

As a future work, we plan to increase the procedural generation capabilities of NOVA, especially regarding the generation of dynamic scene elements other than humans. The feasibility of using physically based rendering will be explored for enhancing the level of provided photorealism. Additionally, we are planning to implement other camera types such as body-worn cameras and third-person-view cameras along with camera artifacts such as motion blur and chromatic aberration to simulate a wider range of real-world video captures. Analyzing the performance of the trackers under various attributes such as different crowdedness levels, weather conditions, times-of-days, environments, and camera setups are crucial for understanding the tracker's performance and the limitations. This analysis that we have done in the work could be extended to more extreme conditions. At the same time, studying the distribution and statistics of scene elements like the number of people and gender distribution in both synthetic and real datasets could be a good future work to understand more the capabilities and limitations of our generated synthetic data. That could be followed by a detailed analysis of how a specific attribute distribution could affect the performance of a specific computer vision model. On the other hand, studying the performance of computer vision models in extreme settings is another research direction. For example, training on one environment (rural environment) and testing on another environment (city with skyscrapers). NOVA currently contains predefined small cities. However, generating cities and buildings procedurally could be an interesting research direction. As another application of NOVA, it could be extended to provide satellite images of these cities. Further possible applications of NOVA is to be adopted for the task of 3D reconstruction given that NOVA can diversify many essential environment aspects and camera parameters for this task.

Chapter 6.

Conclusion

Our work presented a simple yet powerful approach that can handle the current bottleneck of data scarcity. To remedy this issue, we developed a novel rendering engine called NOVA for procedural data generation to be used in various computer vision tasks. NOVA is simple to use tool for creating 3D virtual worlds that can be deployed to generate photorealistic, diverse, and large-scale accurately annotated data. NOVA supports many computer vision tasks including both low and high-level vision problems. The procedural generation concept facilitates the diversity of the generated worlds and human agents. Extensive experiments were performed to prove the usability of the generated synthetic data for both training and testing computer vision algorithms. Visual object tracking was considered in particular and the performance of a large number of state-of-the-art trackers was examined in the scope of this work. By providing a new real-world sequences (PTAW172Real), the poor performance of these trackers under adverse weather conditions was studied and critical observations were reported. We offer a remedy to this problem by generating PTAW217Synth dataset with its diverse and rich training sequences under adverse weather conditions. Our experimental results showed that applying our synthetic sequences for training purposes improves the trackers performance considerably [58, 65].

Acknowledgments

This work was supported in part by GEBIP 2018 Award of the Turkish Academy of Sciences to E. Erdem, BAGEP 2021 Award of the Science Academy to A. Erdem, and by TUBITAK-1001 Program Award No. 217E029.

REFERENCES

[1] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, **2016**.

[2] A Li, M Lin, Y Wu, MH Yang, and S Yan. NUS-PRO: A New Visual Tracking Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):335–349, **2016**.

[3] Y. Wu, J. Lim, and M. Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, **2015**. ISSN 0162-8828. doi:10.1109/TPAMI.2014.2388226.

[4] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, **2015**.

[5] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222. **2017**.

[6] César Roberto De Souza, Adrien Gaidon, Yohann Cabon, and Antonio Manuel López Peña. Procedural generation of videos to train deep action recognition networks. In *CVPR*, pages 2594–2604. **2017**.

[7] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243. **2016**.

[8] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349. **2016**.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*. **2009**.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, **2014**.

[11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, **2009**.

[12] Yuying Ge, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5337–5345. **2019**.

[13] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162. **2018**.

[14] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845. **2017**.

[15] Caner Hazirbas, Lingni Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, **2016**.

[16] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6243–6252. **2017**.

[17] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2443–2451. **2015**.

[18] Jin Kyu Kang, Toan Minh Hoang, and Kang Ryoung Park. Person re-identification between visible and thermal camera images based on deep residual cnn using single input. *IEEE Access*, 7:57972–57984, **2019**.

[19] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, **2016**.

[20] Geoffrey R Taylor, Andrew J Chosak, and Paul C Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, **2007**.

[21] Alireza Shafaei, James J Little, and Mark Schmidt. Play and learn: Using video games to train computer vision models. *arXiv preprint arXiv:1608.01745*, **2016**.

[22] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *European Conference on Computer Vision*, pages 909–916. Springer, **2016**.

[23] Vladimir Haltakov, Christian Unger, and Slobodan Ilic. Framework for generation of synthetic ground truth data for driver assistance applications. In *German Conference on Pattern Recognition*, pages 323–332. Springer, **2013**.

[24] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625. Springer, **2012**.

[25] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, **2011**.

[26] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, **2012**.

[27] Ernest Cheung, Tsan Kwong Wong, Aniket Bera, Xiaogang Wang, and Dinesh Manocha. Lcrowdv: Generating labeled videos for simulation-based crowd behavior learning. In *European Conference on Computer Vision*, pages 709–727. Springer, **2016**.

[28] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint arXiv:1810.08705*, **2018**.

[29] Xuan Li, Kunfeng Wang, Yonglin Tian, Lan Yan, Fang Deng, and Fei-Yue Wang. The paralleleye dataset: A large collection of virtual images for traffic vision research. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13, **2018**.

[30] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126(9):961–972, **2018**.

[31] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117. **2017**.

[32] Igor Barros Barbosa, Marco Cristani, Barbara Caputo, Aleksander Rognhaugen, and Theoharis Theoharis. Looking beyond appearances: Synthetic training data for deep cnns in re-identification. *Computer Vision and Image Understanding*, 167:50–62, **2018**.

[33] Lichao Zhang, Abel Gonzalez-Garcia, Joost van de Weijer, Martin Danelljan, and Fahad Shahbaz Khan. Synthetic data generation for end-to-end thermal infrared tracking. *IEEE Transactions on Image Processing*, 28(4):1837–1850, **2018**.

[34] Unity Multipurpose Avatar System. Uma git repo. `https://github.com/umasteeringgroup/UMA`. Online; accessed: 2019-02-20.

[35] Onur Can Uner, Cem Aslan, Burak Ercan, Tayfun Ates, Ufuk Celikcan, Aykut Erdem, and Erkut Erdem. Synthetic18k: Learning better representations for person re-id and attribute recognition from 1.4 million synthetic images. *Signal Processing: Image Communication*, page 116335, **2021**.

[36] Procedural Worlds. Enviro webpage. Online; accessed: 2019-02-20.

[37] Paul Debevec. Image-based lighting. *IEEE Computer Graphics and Applications*, 22(2):26–34, **2002**.

[38] G. Zaal. HDRI Haven. `https://hdrihaven.com/hdris`. Online; accessed: 2019-02-20.

[39] Unity. Rendering with replaced shaders. `https://docs.unity3d.com/Manual/SL-ShaderReplacement.html`. Online; accessed: 2019-02-20.

[40] Matej Kristan, Jiri Matas, Aleš Leonardis, Tomas Vojir, Roman Pflugfelder, Gustavo Fernandez, Georg Nebehay, Fatih Porikli, and Luka Čehovin. A novel performance evaluation methodology for single-target trackers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2137–2155, **2016**. ISSN 0162-8828. doi:10.1109/TPAMI.2016.2516982.

[41] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Roman Pflugfelder, Joni-Kristian Kamarainen, Luka Cehovin Zajc, Ondrej Drbohlav, Alan Lukezic, Amanda Berg, et al. The seventh visual object tracking vot2019 challenge results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0. **2019**.

[42] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European conference on computer vision*, pages 445–461. Springer, **2016**.

[43] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, **2016**.

[44] Shu Zhang, Elliot Staudt, Tim Faltemier, and Amit K Roy-Chowdhury. A camera network tracking (camnet) dataset and performance baseline. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 365–372. IEEE, **2015**.

[45] Lijun Cao, Weihua Chen, Xiaotang Chen, Shuai Zheng, and Kaiqi Huang. An equalised global graphical model-based approach for multi-camera object tracking. *arXiv preprint arXiv:1502.03532*, **2015**.

[46] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6638–6646. **2017**.

[47] Hamed Kiani Galoogahi, Ashton Fagg, and Simon Lucey. Learning background-aware correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1135–1143. **2017**.

[48] Matthias Mueller, Neil Smith, and Bernard Ghanem. Context-aware correlation filter tracking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2017**.

[49] David S. Bolme, J. Ross Beveridge, Bruce A. Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2010**.

[50] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, **2015**.

[51] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European conference on computer vision*, pages 254–265. Springer, **2014**.

[52] Luca Bertinetto, Jack Valmadre, Stuart Golodetz, Ondrej Miksik, and Philip HS Torr. Staple: Complementary learners for real-time tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1401–1409. **2016**.

[53] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813. **2017**.

[54] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6182–6191. **2019**.

[55] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383. **2019**.

[56] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2019**.

[57] Matthias Muller, Adel Bibi, Silvio Giancola, Salman Alsubaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 300–317. **2018**.

[58] Abdulrahman Kerim, Cem Aslan, Ufuk Celikcan, Erkut Erdem, and Aykut Erdem. Nova: Rendering virtual worlds with humans for computer vision tasks. In *Computer Graphics Forum*. Wiley Online Library.

[59] Microsoft. Microsoft rocketbox avatar library git repo. `https://github.com/microsoft/Microsoft-Rocketbox`. Online; accessed: 2020-05-17.

[60] Abhishek Dutta and Andrew Zisserman. The VIA annotation software for images, audio and video. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19. ACM, New York, NY, USA, **2019**. ISBN 978-1-4503-6889-6/19/10. doi:10.1145/3343031.3350535.

[61] A. Dutta, A. Gupta, and A. Zissermann. VGG image annotator (VIA). http://www.robots.ox.ac.uk/ vgg/software/via/, **2016**. Version: 2.0.10, Accessed: 2020-07-19.

[62] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Atom: Accurate tracking by overlap maximization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4660–4669. **2019**.

[63] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192. **2020**.

[64] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Know your surroundings: Exploiting scene information for object tracking. *arXiv preprint arXiv:2003.11014*, **2020**.

[65] Abdulrahman Kerim, Ufuk Celikcan, Erkut Erdem, and Aykut Erdem. Using synthetic data for person tracking under adverse weather conditions. *Image and Vision Computing*, page 104187, **2021**.